

Vergleich von Verfahren zur Tiefenschätzung in Stereobildpaaren

Diplomarbeit

im Fachbereich Photoingenieurwesen und Medientechnik
an der Fachhochschule Köln

Autor
Sylvia Lange
Köln
Mat.Nr. 1101641211

Referent
Prof. Dr. Dietmar Kunz
Fachhochschule Köln

Koreferent
Prof. Dr. Rudolf Richter
Fachhochschule Köln

Köln, Mai 2005

Comparison of Methods for Depth Estimation in Stereo Image Pairs

Thesis

at the Department of Imaging Sciences and Media Technology
University of Applied Sciences Cologne

Author
Sylvia Lange
Cologne
Mat.Nr. 1101641211

First Reviewer
Prof. Dr. Dietmar Kunz
University of Applied Sciences Cologne

Second Reviewer
Prof. Dr. Rudolf Richter
University of Applied Sciences Cologne

Cologne, May 2005

Danksagung

Mein herzlicher Dank gilt meinem Referenten Prof. Dr. Dietmar Kunz für die viele Zeit und Mühe, die er investiert hat, sowie meinem Koreferenten Prof. Dr. Rudolf Richter.

Inhaltsverzeichnis

1. Einleitung	7
2. Material und Methoden	9
2.1. Winner-Take-All-Verfahren	10
2.1.1. Boxfilter	11
2.1.2. Minfilter	11
2.1.3. Binomialfilter	11
2.1.4. Regulare Diffusion	12
2.1.5. Membrandiffusion	12
2.2. Dynamic Programming und Scanline Optimization	12
2.3. Simulated Annealing	15
2.4. Graph Cuts	17
3. Experimente	19
3.1. Experiment 0: Texturlose Stellen, Diskontinuitäten, Prediction Error	20
3.2. Experiment 1: Lokale Methoden mit quadratischen Fenstern	23
3.3. Experiment 2: Lokale Methoden mit verschiebbaren Fenstern	23
3.4. Experiment 3: Globale Methoden mit verschiedenen Matchingkosten	23
3.5. Experiment 4: Lokale Methoden mit verschiedenen Anhäufungsmethoden	24
3.6. Experiment 5: Globale Methoden mit Variation der optischen Glätte	24
3.7. Experiment 6: Verbesserung der Leistung des Graph Cut Verfahrens	24
3.8. Subpixelverfeinerung	24
4. Berechnung der Disparitäten in selbst erzeugten Stereobildern	26
4.1. Anfertigung von Stereobildern	26
4.2. Berechnung der Helligkeit aus der Entfernung	28
4.3. Fehlerabschätzung	31
4.4. Vergleich des gemessenen mit dem berechneten Abstand	34
4.5. Erstellung von Groundtruth-Karten	35

5. Tests und Ergebnisse	37
5.1. Ermittlung der Leistung der verschiedenen Verfahren und Vergleich mit den Angaben der Autoren	37
5.2. Vergleich des RMS-Fehlers aus Tiefenkarte und Groundtruth-Karte mit dem RMS-Fehler des Prediction Error	40
5.3. Testen der Leistung der Algorithmen an texturlosen Stellen	43
5.4. Optimierung der Parameter	44
5.4.1. Optimierung der Experimente 1 und 2	44
5.4.2. Optimierung von Scanline Optimization	45
5.4.3. Optimierung von Simulated Annealing	46
5.4.4. Dynamic Programming und Graph Cuts	47
5.5. Verkleinerung des Disparitätsschrittes und Subpixelverfeinerung	47
6. Diskussion	49
6.1. Beurteilung der einzelnen Verfahren	49
6.1.1. Lokale Methoden	49
6.1.2. Dynamic Programming	50
6.1.3. Scanline Optimization	50
6.1.4. Simulated Annealing	51
6.1.5. Graph Cuts	51
6.2. Vergleich aller Verfahren	52
6.3. Bewertung des Prediction Error	54
6.4. Bewertung der von den Autoren Scharstein und Szeliski veröffentlichten Bilder und Ergebnisse	54
6.5. Beurteilung der Implementierung	55
7. Zusammenfassung	57

Anhang

A.1.a. Entstehung des Plugins entzerrung_.java	59
A.1.b. Programmcode entzerrung_.java	60
A.2. Vergleich der aus den Tiefenkarten berechneten Entfernungen mit den tatsächlichen Entfernungen, Tabellen	64
A.3. Fehlerauswertung der einzelnen Motive	68
A.4. Endergebnis der Fehlerauswertung: Durchschnittswerte aus allen Motiven	84
A.5. Beste Ergebnisse aus einer Serie: Statistik der Anzahl der besten Ergebnisse in Abhängigkeit von den Parametern	85
A.6. Tabelle der benötigten Zeiten zur Erstellung einer Tiefenkarte	91
A.7. Struktogramm für Simulated Annealing	92
A.8. Lexikon der Abkürzungen	93
A.9. Literaturverzeichnis	94
A.10. Abbildungsverzeichnis	96
A.11. Inhalte der CDs	97

1. Einleitung

Die gleichzeitige Auswertung zweier räumlich versetzter Bilder dient dem Erfassen der dritten Dimension. Wie auch der Mensch nur mit zwei Augen räumliche Tiefe wahrnehmen kann, kann nur mit zwei oder mehr Stereobildern eine Tiefenschätzung möglich werden. Dabei muss einem Objekt in einem Bild das entsprechende Objekt im zweiten Bild zugeordnet werden. Da viele Pixel ähnliche Farb- und Helligkeitswerte haben können oder sich die Farbtöne desselben Objektpunkts durch Schatteneffekte voneinander unterscheiden oder verdeckt werden, ist das so genannte "Korrespondenzproblem" nicht einfach zu lösen.

Laut einem Bericht von Szeliski und Zabih [20] ist der Stand der Dinge bezüglich der Evaluation von Stereomethoden ziemlich dürftig, obwohl eine große Zahl von Algorithmen und Berichten veröffentlicht worden ist. Die meisten Berichte geben jedoch keine Auskunft über quantitative Vergleiche, und wenn, dann beschränken sie sich meist auf synthetische Bilder. Über die Erstellung und Auswertung von Stereobildern wird kaum berichtet.

Stereo Matching ist ein Bereich der Computer Vision, welche ein Teilgebiet der Informatik bezeichnet, das sich mit der Extraktion von Informationen aus Bildern oder Videosequenzen beschäftigt. Die Werkzeuge der Computer Vision stammen meist aus der Mathematik, insbesondere aus der Geometrie, linearer Algebra, Statistik, Optimierung und Funktionsanalysis. Diese Werkzeuge werden in Algorithmen zur Segmentierung und Klassifizierung angewandt [8].

Unter einem Algorithmus versteht man allgemein eine mehr oder weniger genau definierte Handlungsvorschrift zur Lösung eines Problems. Eine einheitliche präzise Definition des Begriffs ist immer noch Gegenstand der Forschung. Eigenschaften eines Algorithmus sind: Das Verfahren muss eindeutig beschreibbar sein. Jeder Schritt muss auch tatsächlich ausführbar sein. Das Verfahren darf nur endlich viele Schritte und für jeden Schritt nur endlich viel Speicherplatz benötigen [26].

Ziel des Stereo Matching ist die Zuordnung zweier Stereobilder zueinander mit dem Ergebnis einer dichten Tiefenkarte. In diesem Bild werden die in den Stereoaufnahmen vorhandenen Objekte in verschiedenen Grautönen wiedergegeben, wobei ein dunkler Ton einen großen Abstand zur Kamera bedeutet, ein heller Ton einen geringen Abstand. Aus den Helligkeiten können die Entfernungen berechnet werden. Der Begriff der dichten ("dense") Tiefenkarte bedeutet in diesem Zusammenhang, dass die Tiefenkarte ein System

voller Werte und mit so wenig Elementen ohne Information ist, dass sie behandelt werden kann, als hätte sie keine solchen Elemente. Somit kann jedem Bildpunkt eine Disparität zugeordnet werden. Das Korrespondenzproblem muss also für jedes Pixel des Referenzbildes gelöst werden. Dafür ist ein Ähnlichkeitsmaß nötig, welches überall im Bild angewandt werden kann. Im Gegensatz dazu enthält eine "sparse" Tiefenkarte nur an relevanten Stellen wie Kanten Informationen.

Einige aktuelle Stereoverfahren sollen in dieser Arbeit mit selbst aufgenommenen Stereobildern getestet und bewertet werden.

Die Aufgabenstellung beinhaltet Internetrecherche, um Berichte über Stereoverfahren sowie Quellcode zu finden, das Kompilieren des Quelltextes und das Testen des Programms. Bei bekannter Brennweite des Kameraobjektivs bei der Aufnahme kann aus dem Helligkeitswert der berechneten Tiefenkarte der Wert des Abstands zurückgewonnen werden und mit dem tatsächlichen, gemessenen Abstand verglichen werden.

Es soll untersucht werden, ob es möglich ist, anhand von zwei Stereobildern Tiefenkarten zu erschaffen, die die Voraussetzungen zur 3D-Rekonstruktion der Szene erfüllen. Da innerhalb eines Referenzbildes die x- und die y-Koordinate eines jeden Bildpunktes bekannt sind und aus der Tiefenkarte die z-Komponente, also die Entfernung ermittelt werden kann, liegen für jedes Pixel alle Koordinaten für eine räumliche Darstellung vor. Der Schwerpunkt der vorliegenden Arbeit liegt somit auf experimentellen Tests und Auswertungen mit eigenen Bildern.

Für die Berechnungen muss der Begriff der Disparität definiert werden. Die Disparität ist der Positionsunterschied der jeweiligen Gegenstände in den Stereobildern. Sie kann sowohl in Pixeln, als auch in cm angegeben werden. Die Disparität von entfernt liegenden Objekten in einem Bild ist kleiner als die von nahe gelegenen Objekten. Die Disparität ist vom Abbildungsmaßstab und von der Kameraverschiebung abhängig.

2. Material und Methoden

Auf der Internetseite www.middlebury.edu/stereo ist sowohl ein Bericht über eine Zusammenstellung von Stereoverfahren zu finden, als auch C++ Quelltext, welcher die Grundlage der weiteren Untersuchungen bildet [1].

Das Programm basiert auf einer Implementierung von fünf verschiedenen Verfahren und ist ausbaufähig, das heißt, weitere Verfahren können eingebaut werden. Alle Verfahren basieren auf mehreren Teilschritten, deren jeweilige Parameter veränderbar sind und deren Auswirkungen in verschiedenen Experimenten getestet werden können.

Es muss zwischen lokalen und globalen Verfahren unterschieden werden: Lokale Verfahren versuchen mit Blockmatching in beiden Bildern ähnliche Stellen zu finden, während globale Verfahren alle Bildpunkte bearbeiten, um schrittweise eine ideale Lösung zu finden. Bei einem lokalen Optimierungsproblem beschränkt man sich darauf, eine lokale Lösung des vorgegebenen Problems zu finden. Möchte man wirklich *die Lösung* finden, so hat man ein globales Optimierungsproblem vor sich. Lokale und globale Optimierungsprobleme sind von stark unterschiedlicher Komplexität [3]. Während fast alle lokalen Optimierungsprobleme in polynomialer Laufzeit gelöst werden können, sind globale Probleme außer in einfachen Spezialfällen NP-hart, benötigen also exponentiellen Aufwand.

In lokalen Methoden wird der Disparitätswert jedes Pixels unabhängig von anderen Pixeln ausgewählt, globale Algorithmen versuchen das Problem zu lösen, indem sie eine bestimmte Energiefunktion minimieren. Die Energiefunktion ist die Kombination eines Datenterms und eines Glattheitsterms. Der Datenterm misst, wie gut die Disparitätsfunktion mit dem Input-Bildpaar übereinstimmt. Der Glattheitsterm codiert die Glattheitsannahmen, die der Algorithmus macht. Oft werden dafür nur die Differenzen zwischen den Disparitäten benachbarter Pixel gemessen. Der Hauptunterschied zwischen den globalen Verfahren besteht in der Vorgehensweise zur Energieminimierung. Im Allgemeinen arbeiten lokale Algorithmen schneller, globale geben jedoch die besseren Ergebnisse.

Die implementierten Algorithmen heißen Winner-Take-All, Dynamic Programming, Scanline Optimization, Simulated Annealing und Graph Cuts. Außer dem ersten sind alle Verfahren global.

Jeder Stereoalgorithmus leistet im Allgemeinen vier Schritte: Matchingkostenberechnung, Kostenanhäufung, Disparitätsberechnung und Disparitätsverfeinerung. Dabei liegt der Schwerpunkt bei den lokalen Methoden auf der Kostenberechnung, während globale Algorithmen die meiste Arbeit in die Anhäufungsphase investieren.

In der Implementierung von Daniel Scharstein und Richard Szeliski wird ein Algorithmus nicht als Black Box behandelt, sondern jeder der einzelnen Schritte wird in Experimenten untersucht und die Komponenten variiert.

2.1. Winner-Take-All-Verfahren

Die Vorgehensweise im Winner-Take-All-Verfahren besteht aus den Teilschritten Matchingkostenberechnung, Aggregation / Anhäufung und Disparitätsberechnung. Unter Matchingkosten versteht man bestimmte Werte, die berechnet werden, um Informationen über die Ähnlichkeit der Blöcke im linken und im rechten Bild zu erhalten. Diese Werte sind z.B. die Summe der quadrierten Differenzen der Pixel zweier Blöcke, die jeweils um eine bestimmte Disparität, die nicht größer als d_{\max} , also die maximale im Bild vorkommende Disparität sein muss, gegeneinander verschoben sind.

In Abbildung 1a ist die Vorgehensweise des Verschiebens dargestellt, wobei die Fenstergröße zur besseren Darstellung verhältnismäßig zu groß ist. Je ähnlicher sich zwei Blöcke sind, umso kleiner werden die Matchingkosten. Anhäufung wird praktiziert, indem Summen und Mittelwerte über eine bestimmte Region gebildet werden. Das Berechnen der endgültigen Disparitäten ist in diesem Fall einfach: Für jedes Pixel muss die Disparität d ausgewählt werden, die mit dem geringsten Kostenwert verbunden ist (Abb. 1b).

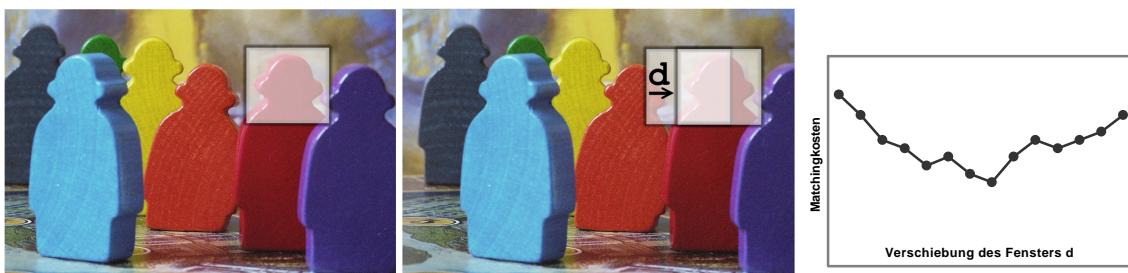


Abb. 1a: Verschiebung des Fensters entlang einer Scanlinie zur Gewinnung der Matchingkosten für verschiedene Werte von d [A9]

Abb. 1b: Auswählen der Disparität mit den geringsten Kosten [A9]

Bei iterativen Verfahren werden mehrfach zu den Matchingkosten eines Pixels die gewichteten Matchingkosten der Nachbapixel addiert.

Ein zentrales Problem der lokalen Methoden ist das Finden der optimalen Fenstergröße. Wenn das Fenster zu klein ist, könnte aufgrund von Rauschen und Mehrdeutigkeiten eine

falsche Zuordnung gefunden werden, wenn es zu groß ist, kann das untersuchte Gebiet nicht als Ganzes einer Disparität zugeordnet werden. Problemzonen wie Stellen mit perspektivischer Verkürzung oder verborgene Stellen führen zum Verlust von Details, zu einer unscharfen Wiedergabe oder falscher Positionierung der Objektgrenzen.

Werden Punkte, die beispielsweise im Eckbereich eines Vordergrundobjektes liegen, von einem Fenster erfasst, so liegen mehr als die Hälfte der vom Fenster abgedeckten Fläche im Hintergrund. Den Punkten wird folglich eine falsche Disparität zugeordnet. Der allgemeine Effekt ist die Abrundung von Ecken, weil Punkte in der Nähe von Ecken in die falsche Disparität übernommen werden. Geradlinige Disparitätskanten sind von diesem Problem nicht betroffen. Kanade und Okutomi [22] schlagen als Lösung adaptive Fenster vor. Die optimale Fenstergröße wird durch Einschätzen der Disparitätsunsicherheit im Fenster gefunden.

Scharstein und Szeliski erwähnen in [1] einige weitere Möglichkeiten und Varianten der Anhäufung. In den implementierten Versuchen wird Anhäufung mit einem der folgenden Filtern bzw. Verfahren praktiziert:

2.1.1. Boxfilter

Mit einem Boxfilter, der in die Gruppe der Glättungsfilter gehört, werden die Bildpunkte in einem kleinen Bildbereich gemittelt. Im Prinzip werden alle Bildpunkte innerhalb der Filtermaske addiert und die Summe durch die Anzahl der Pixel geteilt.

2.1.2. Minfilter

Ein Minfilter selektiert aus allen Pixeln innerhalb der Filtermaske das Pixel mit dem geringsten Wert und schreibt ihn in das zentrale Pixel.

2.1.3. Binomialfilter

Ein Binomialfilter ist ein Glättungsfilter, dessen Filtermaske Werte einer Reihe des Pascalschen Dreiecks enthält, in diesem Fall die Binomialkoeffizienten 1, 4, 6, 4, 1, laut der Steuerungstextdatei zu Experiment 4.

2.1.4. Reguläre Diffusion

Ein anderer Ansatz ist die Anhäufung mit einem ungleichmäßigen Diffusionsprozess nach Scharstein, Szeliski [21]. Anstelle eines Fensters kann mit einer gewichteten Hilfsfunktion wie mit der Gaußkurve iterativ angehäuft werden. Eine begrenzte Anzahl von einfachen Diffusionsschritten ergibt Resultate, die ähnlich denen der quadratischen Fenster sind. Vorteilhaft wirkt sich die Rotationssymmetrie der Funktion aus, sowie die Tatsache, dass entfernte Punkte weniger Einfluss haben.

2.1.5. Membrandiffusion

Ein Nachteil der einfachen Diffusion ist, dass die Größe der untersuchten Region mit der Zahl der Iterationen steigt. Während die Diffusion zu einem bestimmten Wert, der schließlich das ganze Bild bedeckt, konvergiert, besteht das Interesse, einen vermittelnden Zeitpunkt zu finden, bei dem die Diffusion nur bis zu einem bestimmten Punkt fortgeschritten ist und das bestmögliche Ergebnis liefert.

Dafür wird der Diffusionsgleichung ein Term hinzugefügt, der den Betrag misst, um den der jeweilige Wert vom originalen Wert abweicht. Diese Gleichung wird Membrangleichung genannt. Das Benutzen des Membranmodells erleichtert das Problem des Konturverschimmens etwas.

2.2. Dynamic Programming und Scanline Optimization

Dynamic Programming kann in kurzer Zeit das globale Minimum für unabhängige Scanlinien finden. Die Matchingkosten zwischen je zwei korrespondierenden Scanlines werden in eine Matrix eingetragen, mit der der Pfad der Minimalkosten berechnet wird. Zum Verständnis des Dynamic Programming und aller anderen globalen Methoden muss der Begriff des Disparitäts-Raum-Bildes eingeführt werden. Ein Disparitäts-Raum-Bild stellt die Differenzen der Pixel zweier Stereobilder dar. Dabei entsteht ein dreidimensionales Bild entlang der Koordinaten x , y und d (Disparität). Das Bild erstreckt sich somit über einen Raum, der von $x = 0$ bis $x = \text{Weite} - 1$, $y = 0$, $y = \text{Höhe} - 1$ und d_{\min} bis d_{\max} reicht. Da eine räumliche Anordnung von Pixeln visuell nicht erfassbar ist, kann ein x - d -Schnitt für ein konstantes y oder ein x - y -Schnitt für ein bestimmtes d angefertigt werden. Eine Darstellung dieser Schnitte liefert Abbildung 3c.

Nach Thiel [9] ist das Vorgehen für jeweils eine Scanlinie des linken und des rechten Bildes wie folgt: Zum Erstellen der nullten Zeile eines x-d-Schnittes eines DSI (disparity space image) werden beide Scanlinien einer bestimmten Höhe y mit dem Versatz 0 entsprechend der Disparität 0 übereinander gelegt und die Differenz der Pixel in das DSI geschrieben. Für die nächste Zeile wird eine Zeile gegen die andere um ein Pixel verschoben und die Differenzen berechnet. Das DSI zeigt also die Übereinstimmungen beider Bilder in Abhängigkeit von der Disparität.

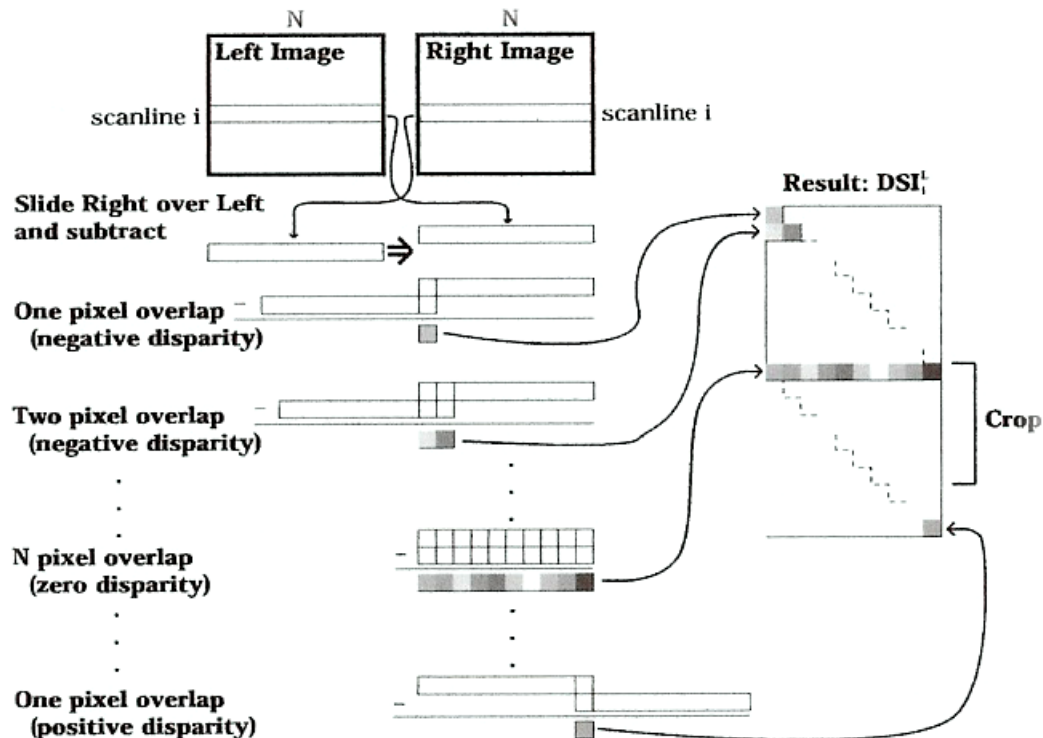


Abb. 2: Vorgehensweise zur Erstellung eines x-d-Schnittes eines DSI für die Scanlinie i [A1]

Um das gesamte Bild zu analysieren, muss jedes Scanlinienpaar auf diese Weise bearbeitet werden. Wenn eine Region mit ausgeprägter Oberflächenstruktur einer Scanlinie über die entsprechende Region der anderen Scanlinie gezogen wird, so kann man im DSI eine Linie mit dunklen Pixeln sehen. Werden dagegen zwei strukturarme korrespondierende Regionen übereinander gezogen, so wird eine diamantenförmige schwarze Region sichtbar.

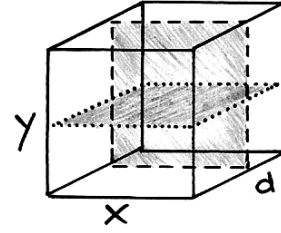
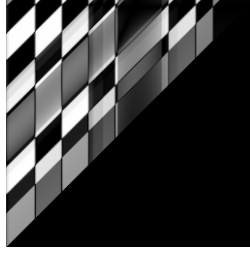


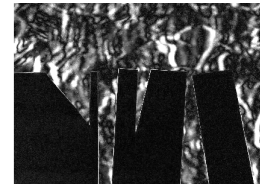
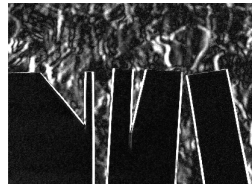
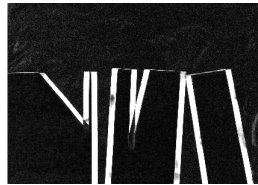
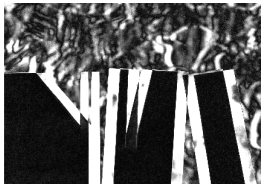
Abb. 3a: linkes Stereobild "Corridor" [A2]

Abb. 3b: x-d-Schnitt durch das DSI des Motivs "Corridor" [A9]

Abb. 3c: Schnitte durch das DSI: = x-d-Schnitt; ----- = x-y-Schnitt [A9]

Anhand der Abbildungen wird das Prinzip des Übereinanderziehens deutlich. Bild 3b zeigt den x-d-Schnitt der Bildzeile 213 am unteren Rand des Bildes 3a und seinem korrespondierenden Stereobild. Es wurden absolute Differenzen als Matchingkosten benutzt. Relevant ist nur der kleine Bereich zwischen d_{\min} und d_{\max} , also in diesem Beispiel etwa die obersten 20 Pixel des DSI-Schnittes.

Durch die Auswertung von x-y-Schnitten durch ein DSI kann bestimmten Bildteilen eine bestimmte Disparität zugeordnet werden. In der folgenden Serie mit zunehmendem d des Motivs Papier gibt das zweite Bild die Disparität des Hintergrundes wieder, das vierte die Disparität des Vordergrundes.

Abb. 4 a, b, c, d: x-y-Schnitte durch das DSI des Motivs Papier für verschiedene Werte von d [A9]

Der "Null-Pfad" durch das DSI, der zu der Verschiebung in Beziehung steht und die Höhenlinie widerspiegelt, kann als schwarze Linie in der x-d-Ansicht gefunden werden. Je nach Art der Bilder ist die Linie für den Betrachter nicht einfach zu erkennen.

Der Disparitätsraum $E(x, y, d)$ wird von Scharstein und Szeliski in [21] beschrieben. Der noch unbearbeitete Disparitätsraum E_0 wird definiert als

$$E_0(x, y, d) = \mathbf{r}(I_L(x + d, y) - I_R(x, y)) \quad (1)$$

I_L und I_R sind die Intensitätsfunktionen des linken und des rechten Bildes und \mathbf{p} eine Funktion, die die Ähnlichkeiten beider Intensitäten misst, beispielsweise anhand der quadrierten Differenzen:

$$\mathbf{r}(l - r) = (l - r)^2 \quad (2)$$

Diese Formulierungen benutzen I_R als Referenzbild. Nach der Anhäufung in einem endgültigen Raum $E(x, y, d)$ kann die Disparitätsfunktion wie folgt definiert werden:

$$d(x, y) = \arg \min E(x, y, d) \quad (3)$$

Das Prinzip aller globalen Methoden besteht darin, diese Funktion zu minimieren.

Mit Dynamischer Programmierung soll für jede einzelne Scanlinie der beste Pfad durch das DSI gefunden werden. Dabei wird anstelle eines Disparitäts-Raum-Bildes mit einer Matrix, in die die Differenzwerte geschrieben werden, gearbeitet. Der Algorithmus minimiert die Kosten des Pfades, die dem Absolutwert des DSI-Pixels an dieser Stelle entsprechen. Da jedes Scanlinienpaar einzeln bearbeitet wird, entsteht in der Tiefenkarte eine typische streifenförmige Struktur.

Scanline Optimization arbeitet ähnlich wie Dynamic Programming an verschiedenen x-d-Schnitten und optimiert jede Scanlinie einzeln. Dabei werden verschiedene Glattheitsterme benutzt. Jedem Punkt x wird ein Wert d zugewiesen, so dass die Kosten entlang der Scanlinie minimiert werden. Ohne den Glattheitsterm entspricht dies dem Prinzip von WTA. Der SO-Algorithmus löst das gleiche Optimierungsproblem wie der in Kapitel 2.4. beschriebene Graph Cut Algorithmus, mit dem Unterschied, dass vertikale Glattheitsterme ignoriert werden.

Da Dynamic Programming und Scanline Optimization jede Linie der Stereobilder unabhängig bearbeiten, sind sie fehleranfällig in Situationen, in denen sich zwei zusammengehörende Punkte in zwei verschiedenen Scanlinien befinden. Dies äußert sich in der Disparitätskarte in horizontalen "streaks", Streifen, Spuren.

2.3. Simulated Annealing

Simulated Annealing ist ein Suchverfahren mit dem Ziel der Energieminimierung einer Funktion. Das Prinzip besteht darin, von einem gegebenen Punkt im Lösungsraum zu einem benachbarten überzugehen und die dabei entstehende Veränderung unter bestimmten Bedingungen zu akzeptieren. Eine Verbesserung wird immer akzeptiert, eine Verschlechterung manchmal, besonders in der frühen Phase der Suche, abhängig von den Parametern.

Der Begriff "Simulated Annealing", also simulierte Abkühlung begründet sich auf die thermodynamische Betrachtung von abkühlendem Metall, welches Kristalle bildet, die im Idealfall eine möglichst günstige Energiebilanz besitzen. Bei langsamem Abkühlen können sich die Moleküle ordnen und bessere energetisch stabile Strukturen bilden. Zu schnelles Abkühlen erreicht nicht das globale Minimum. Sollte ein lokales Minimum erreicht

werden, das jedoch nicht das globale ist, kann das Verfahren durch vorübergehende Verschlechterung das Hindernis überwinden und die Ideallösung finden [4], [5], [6].

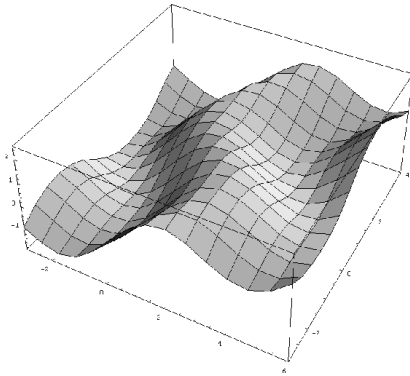


Abb. 5a: Beispiel einer Zielfunktion (Wertelandschaft) [A7]

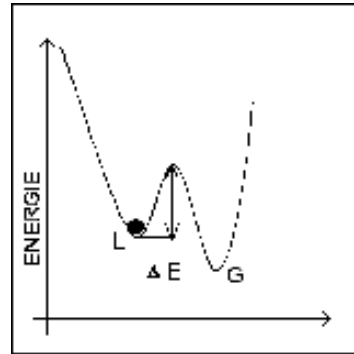


Abb. 5b: Lokales und globales Minimum [A6]

Beim Simulated Annealing benötigt man eine Kostenfunktion, die die Qualität eines durchgeführten Schrittes bewertet. Man erzeugt eine zufällige Ausgangskonfiguration und untersucht dann benachbarte Lösungen bezüglich der Kostenfunktion. Ob die neue Konfiguration als Ausgangspunkt für einen neuen Suchlauf akzeptiert wird, hängt von der Qualität der Konfiguration und einem Kontrollparameter ab. Befindet man sich an einem Ort x und möchte zu Ort y wechseln, muss die Energiedifferenz berechnet werden:

$$\Delta E = f(y) - f(x) \quad (4)$$

Ist $\Delta E < 0$, so wird y als neuer Ort akzeptiert. Andernfalls wird y als neuer Ort nur dann akzeptiert, wenn eine gerade ermittelte Zufallszahl zwischen 0 und 1 kleiner als

$$\exp\left(-\frac{\Delta E}{T}\right) \text{ ist.} \quad (5)$$

Da die Temperatur T langsam sinkt, werden anfangs fast alle neuen Konfigurationen akzeptiert, während zum Schluss nur noch kleine Änderungen zugelassen werden. Ein Struktogramm der Vorgehensweise des Simulated Annealing findet sich im Anhang A.7.

Simulated Annealing gilt als meta-heuristisches Verfahren, da es im Gegensatz zu heuristischen Verfahren vorübergehende Verschlechterungen akzeptiert, um aus einem lokalen Minimum heraus zu einem globalen Minimum zu finden. Detailveränderungen zum Schlechteren sind zwischendurch notwendig, um das Gesamtsystem zu verbessern und ein neues Gleichgewicht zu finden [26].

2.4. Graph Cuts

Graph Cuts sind wörtlich Schnitte durch einen Graph. Ein Graph ist ein netzähnliches Modell, welches aus einer Anzahl Knoten und einer Anzahl Verbindungen zwischen den Knoten besteht. Die Knoten entsprechen in diesem Fall Pixeln.

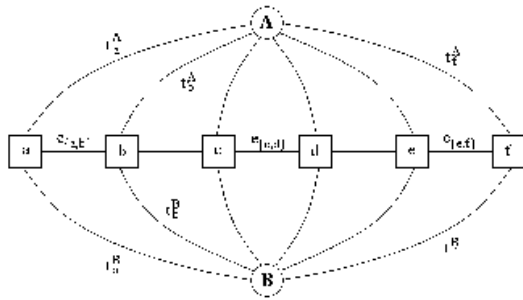


Abb. 6a: Graph [A3]

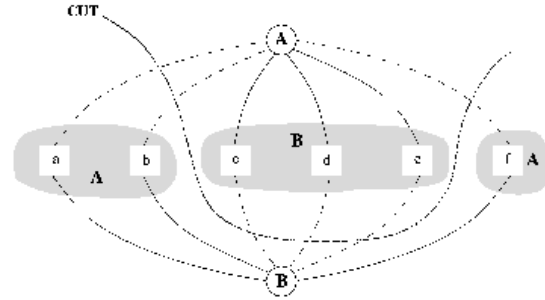


Abb. 6b: Schnitt durch einen Graph [A3]

Alle Kanten sind mit Matchingkosten, die aus dem DSI stammen, belegt. Das Ziel ist, Schnitte durch den Graph zu legen, sodass die Energiefunktion minimiert wird. Auf diese Weise kann jedes Pixel einen Disparitätswert erhalten, der in einen Helligkeitswert umgewandelt wird. Pro Durchgang findet eine Aufteilung der Pixel in diejenigen, die die alte Disparität behalten, und jene, die die nächste Disparität bekommen, statt. Jedem Pixel muss eines von zwei Labeln mit der jeweiligen Disparität zugeordnet werden, im Bild A und B genannt.

Die Label-Paarungen für jede einzelne Iteration werden jeweils zufällig zusammengestellt, und das Verfahren beendet, wenn keine weiteren Energieverbesserungen mehr möglich sind. Anhand des folgenden Ausschnitts der Berichterstattung kann die Vorgehensweise zur Energieminimierung nachvollzogen werden:

```
- optimizing: Graph Cut (near global minimum)
graph cut: alpha=15, beta=3
  old E = 10476451.00 (6336211.00 + 4140240.00)
  new E = 10471911.00 (6337071.00 + 4134840.00)
graph cut: alpha=9, beta=5
  old E = 10471911.00 (6337071.00 + 4134840.00)
  new E = 10454008.00 (6341128.00 + 4112880.00)
graph cut: alpha=2, beta=1
  old E = 10454008.00 (6341128.00 + 4112880.00)
  new E = 10390323.00 (6348063.00 + 4042260.00)
graph cut: alpha=4, beta=0
  old E = 10390323.00 (6348063.00 + 4042260.00)
  new E = 10361913.00 (6352173.00 + 4009740.00)
...
```

Die Label stellen besondere Knoten im Graph dar, Quelle und Senke. Alle Verbindungen der Pixel zu den Labeln enthalten Energiewerte. Die Verbindung mit der jeweils kleineren Energie wird durchtrennt und dem Pixel das Label der durchtrennten Seite zugewiesen. Ein

Cut ist somit die Partition der Knoten in zwei Teilmengen. Für jedes benachbarte Pixelpaar bestehen ebenfalls Verbindungen, welche mit einem bestimmten Penalty-Wert belegt sind. Dieser Wert ist ein "Strafmaß" für die Zuordnung verschiedener Disparitätslevel für die angrenzenden Pixel.

Der maximale Fluss "max flow" ist bildlich z.B. die maximale Menge Wasser, die von der Quelle zur Senke fließen könnte, wenn die Verbindungen des Graphs als "Wasserleitungen" interpretiert würden, deren Kapazitäten gleich der Wichtung wären. Ein Minimal-Schnitt "min cut" durch den Graph bedeutet, dass beim Zuordnen der Labels die Energie minimiert wurde [10], [11], [13]. Der im Programm verwendete Graph Cut Algorithmus stammt von der Internetseite [2].

Der Algorithmus kann durch α - β -Tauschbewegungen verbessert werden. Für ein festgesetztes Paar von Labels α und β werden innerhalb einer Untergruppe von Pixeln der Gruppe α und einer angrenzenden Gruppe mit dem Label β die Label ausgetauscht. Durch Iteration über alle verschiedenen Paare α , β erlaubt diese Technik den steilsten Abstieg bei der Suche nach dem Minimum. Die Eigenschaften eines solchen Algorithmus hängen sehr stark von der Intensität der Tauschbewegung ab.

3. Experimente

Die Implementierung von Scharstein und Szeliski bietet eine Anzahl von vorgegebenen Experimenten, die Serien von Bildern mit veränderten Parametern berechnen.

Das Programm arbeitet mit Scripten, d.h. mit Textdateien, die der Steuerung dienen. Die Namen der zu matchenden Bilder werden in `data_in.txt` geschrieben, die weiteren Parameter in `param_in.txt`.

Die Eingabe in die Konsole lautet: `StereoMatch script exp_all.txt`, wobei diese Textdatei der Reihe nach die einzelnen Ordner der verschiedenen Motive aufruft. Mit der Datei `exp_dispatch.txt` wird das entsprechende Experiment ausgewählt und wiederum die entsprechende Datei aufgerufen. Es besteht die Möglichkeit, Serien von Bildern in Kombination mit verschiedensten Parametern rechnen zu lassen, bestimmte Kombinationen, die jedoch erfahrungsgemäß keine guten Ergebnisse liefern, können in der Datei einfach auskommentiert werden, d.h. ein `#` am Zeilenanfang verhindert die Bearbeitung dieser Zeile. Alle Bezeichnungen für Parameter, Verfahren oder Fehlermaße wurden aus dem Bericht von Scharstein und Szeliski [1] übernommen und können gegebenenfalls dort nachgelesen werden.

Für eine quantitative Fehlerauswertung wird neben mindestens zwei Stereobildern eine Groundtruth-Karte benötigt, welche eine ideale Tiefenkarte darstellt, also Auskunft über die tatsächlichen Disparitäten des Motivs gibt. Durch den Vergleich mit einer von einem der Verfahren erzeugten Tiefenkarte können verschiedene Fehlerwerte berechnet werden. Wichtige Anhaltspunkte zur Leistung der Algorithmen liefern drei Maße, die prozentuale Fehler in bestimmten Bildteilen angeben: Der Prozentsatz der schlechten Pixel in nicht verborgenen Gebieten, der Prozentsatz der schlechten Pixel in texturlosen Gebieten und der Prozentsatz der schlechten Pixel in Gebieten nahe Tiefendiskontinuitäten. Als schlechte Pixel gelten in diesem Fall Bildpunkte, die mit einer Disparitätsfehlertoleranz δ_d größer 1 falsch zugeordnet werden.

$$B = \frac{1}{N} \sum_{(x,y)} (|d_C(x, y) - d_T(x, y)| > \delta_d) \quad (6)$$

d_C steht für die berechnete Tiefenkarte, d_T für die Groundtruth-Karte.

Ein weiteres Fehlermaß ist der RMS-Fehler, der den Fehler zwischen den berechneten Disparitätskarten und der Groundtruth-Karte in Disparitätseinheiten wiedergibt:

$$R = \left(\frac{1}{N} \sum_{(x,y)} |d_C(x,y) - d_T(x,y)|^2 \right)^{\frac{1}{2}} \quad (7)$$

N steht in diesem Fall für die Zahl der Pixel.

Der Fehler ergibt sich aus der Wurzel des Mittelwertes der quadrierten Differenz von je zwei korrespondierenden Pixeln und ist umso größer, je schlechter die Qualität des Bildes ist.

Während der prozentuale Fehler nur zwischen richtig und falsch zugeordneten Werten unterscheidet, geht in den RMS-Fehler auch die Größe der Abweichung ein. Daher liefert der Prozentsatz der falsch zugeordneten Pixel eine gute Indikation der Allgemeinleistung eines Algorithmus, während der RMS-Fehler von den möglicherweise sehr großen Disparitätsfehlern einiger weniger Bildstellen verunreinigt werden könnte. Das Programm erstellt auf Wunsch Bilder, in denen falsch gematchte Pixel, bzw. die Größe des Fehlers sichtbar gemacht werden. So kann festgestellt werden, welche Bildbereiche besonders fehleranfällig sind. Dafür muss die Variable `eval_error_scale` in der jeweiligen Textdatei des entsprechenden Experiments auf 1 gesetzt werden.

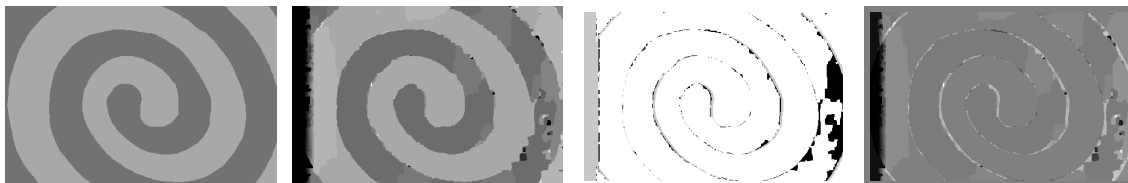


Abb. 7a: Groundtruth-Karte für das Motiv Chaos_flach

Abb. 7b: Tiefenkarte, erstellt mit Graph Cuts

Abb. 7c: Verteilung von richtig zugeordneten Pixeln (weiß) und falschen Pixeln (schwarz)

Abb. 7d: Darstellung der Größe des Fehlers als Abweichung vom mittleren Grauton

Abb. 7 a – d: [A9]

Anhand dieses Beispiels wird sichtbar, dass hier Schwierigkeiten besonders in Tiefendiskontinuitäten, also am Übergang von Vordergrund zu Hintergrund auftreten.

3.1. Experiment 0

Texturlose und verborgene Stellen, Diskontinuitäten, Prediction Error

Dieses Experiment analysiert die Eigenschaften von Stereobildern. Es werden keine Tiefenkarten erstellt, stattdessen entstehen Bilder mit der Darstellung der Verteilung von texturlosen und verborgenen Regionen im Bild, sowie der Abbildung von Diskontinuitäten.

Ein Pixel stellt eine Tiefendiskontinuität dar, wenn eines der vier benachbarten Pixel in der Groundtruth-Karte eine Disparität hat, die um mehr als 1 von der eigenen Disparität abweicht. Ein Pixel ist verborgen, wenn es im zu matchenden Bild nicht sichtbar ist. Texturlose Stellen können nicht aus der Tiefenkarte abgelesen werden. Stattdessen wird nach größeren Gebieten gesucht, in denen der Bildgradient klein ist: Der quadrierte horizontale Intensitätsgradient, gemittelt über ein Fenster bestimmter Größe, muss unter einem vorgegebenen Grenzwert liegen.



Abb. 8a: Tiefendiskontinuitäten

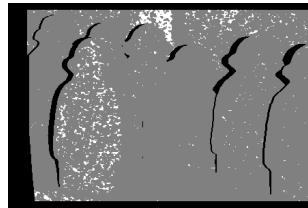


Abb. 8b: Texturlose Stellen



Abb. 8c: Verborgene Pixel

8 a, b, c: [A9]

Bild 8a zeigt Tiefendiskontinuitäten des Motivs Spiel als weiße Umrisse. Die Breite der Diskontinuitätsregion wird von der Variablen `eval_discont_width` bestimmt und beträgt im Allgemeinen 9 Pixel. Bild 8b zeigt texturlose Stellen als weiße Flecken. Diese Stellen werden ausgewertet, um den Prozentsatz der falsch zugeordneten Bildpunkte in texturlosen Bereichen zu erhalten. Bild 8c zeigt verborgene Pixel.

Ein Versuch, die Qualität eines Stereoalgorithmus zu messen, ist, zu testen, wie gut anhand einer Tiefenkarte neue Bildansichten vorhergesagt werden können. Dabei wird aus der Tiefenkarte die Disparität jedes Pixels entnommen und der entsprechende Bildpunkt eines Stereobildes um diesen Wert verschoben. So entsteht ein Bild, das im Idealfall dem zweiten Stereobild entspricht. Durch die Verschiebung entstehen Risse an verborgenen Stellen, welche farbig ausgefüllt werden und nicht in die Bewertung eingehen.

Ziel der Bildvorhersage ist es, Pixel in zwei oder mehr Bildern in einer bestimmten Beziehung zueinander zu platzieren, um eine dichte Beschreibung einer neuen Ansicht zu erhalten. Um die Vorgehensweise zu verdeutlichen, wurde die Groundtruth-Karte des Motivs Spiel, Abb. 9a, benutzt, um einen senkrechten Farbverlauf zu bearbeiten. Jede Farbfläche wird um eine bestimmte Strecke verschoben. Die Figuren in Abb. 9b wirken wie ausgeschnitten und versetzt wieder aufgeklebt.

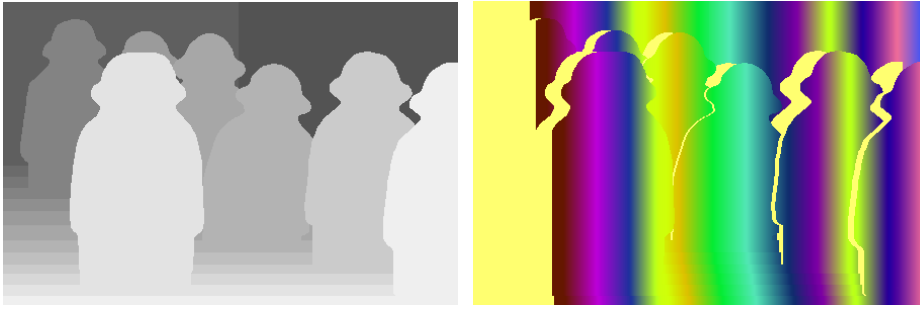


Abb. 9a: Groundtruth-Karte des Motivs Spiel [A9]

Abb. 9b: Inverse Warping: Verschieben der Pixel um den Wert der Disparität am Beispiel eines Farbverlaufs [A9]

Aus dem Vergleich des Ergebnisses mit dem echten Stereobild kann der RMS-Fehler ermittelt werden, der als quantitative Abschätzung des Prediction Error dient. Der eindeutige Vorteil des Prediction Error ist, dass keine Groundtruth-Daten benötigt werden. Die Bildvorhersage kann nach zwei verschiedenen Prinzipien erfolgen: Forward Warping und Inverse Warping. Beim Forward Warping werden aus einem Referenzbild eine oder mehrere neue Ansichten erschaffen, während beim Inverse Warping aus den anderen Bildern jeweils das Referenzbild errechnet wird. Für die folgenden Beispiele wurde eine mit dem Graph-Cut-Verfahren erstellte Tiefenkarte benutzt.

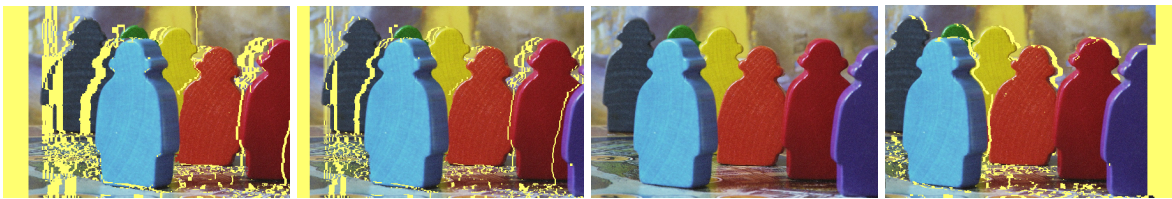


Abb. 10 a, b, c, d: Forward Warping [A9]



Abb. 11 a, b, c, d: Inverse Warping [A9]

Der Prediction Error ist tolerant in einfarbigen Flächen und empfindlich bezüglich farblich unpassender Pixel. Damit nimmt er Bildstörungen ähnlich wie das menschliche Auge wahr. Wird beispielsweise ein blaues Pixel falsch gematcht, jedoch in ein Gebiet eines monochrom blauen Bereichs des Vergleichsbildes, beeinflusst es den Prediction Error nicht negativ. Daher stellt sich die Frage, ob der RMS-Fehler, der aus Groundtruth und Tiefenkarte berechnet wird, ein gutes Maß der Nützlichkeit eines Algorithmus ist, und ob es wirklich nötig ist, in schwach texturierten Gebieten, die schwer zu bearbeiten sind, genauso akkurat zu sein, wie in strukturierten Bereichen. Ist es also besser, gut aussehende

Tiefenkarten ohne Artefakte zu erstellen, oder gute Leistung durch exakte Vorhersage der Szene zu bringen?

Einzelheiten zum Prediction Error werden beschrieben in [19] und [20].

Alle weiteren Experimente können mit oder ohne Groundtruth-Karte durchgeführt werden. Jedoch ist eine quantitative Fehlerabschätzung nur mit der Karte möglich.

3.2. Experiment 1

Lokale Methoden mit quadratischen Fenstern

In Experiment 1 werden verschiedene Matchingkosten unter Verwendung eines lokalen Algorithmus und eines 9×9 Fensters zur Anhäufung getestet. Die Matchingkosten bestehen aus absoluten und quadrierten Differenzen. Sie werden mit verschiedenen Parametern, so genannten "Begrenzungswerten" (Truncation) und dem Verfahren von Birchfeld und Tomasi kombiniert. Dieses Verfahren vergleicht jedes Pixel des Referenzbildes mit einer linear interpolierten Intensitätsfunktion des Matchingbildes und wählt daraus den kleinsten quadratischen Fehler als Matchingkosten aus [12].

Durch die Truncation-Werte wird das Matching-Ergebnis auf einen festgelegten Maximalwert beschränkt.

3.3. Experiment 2

Lokale Methoden mit verschiebbaren Fenstern

Experiment 2 arbeitet genauso wie das erste, jedoch wird ein 9×9 Min-Filter benutzt. Die Wirkung eines Min-Filters kann auch mit verschiebbaren Fenstern erzielt werden.

3.4. Experiment 3

Globale Methoden mit verschiedenen Matchingkosten

In Experiment 3 wird untersucht, wie die Matchingkosten, hier absolute Differenzen, globale Algorithmen beeinflussen. Als Optimierungstechniken dienen Dynamic Programming, Scanline Optimization und Graph Cuts. Es werden wie in den Experimenten 1 und 2 verschiedene Werte der Truncation benutzt, teilweise in Kombination mit dem Birchfeld-Tomasi-Verfahren.

3.5. Experiment 4

Lokale Methoden mit verschiedenen Anhäufungsmethoden

In Experiment 4 werden verschiedene Anhäufungsmethoden, die mit lokalen Methoden benutzt werden, miteinander verglichen. Als Matchingkosten werden absolute Differenzen ohne Truncation benutzt und mit der Winner-Take-All-Methode optimiert.

Fünf verschiedene Anhäufungsmethoden wurden implementiert: Quadratische Fenster mit Fenstergröße 3 bis 29, Verschiebbare quadratische Fenster (Minfilter) mit Fenstergröße 3 bis 29, iterative Binomialfilter für bis zu 28 Iterationen, Reguläre Diffusion für bis zu 150 Iterationen und Membrandiffusion für 150 Iterationen mit verschiedener Gewichtung der Originalkostenwerte in der Diffusionsgleichung.

3.6. Experiment 5

Globale Methoden mit Variation der "optischen Glätte"

Experiment 5 erforscht die vier globalen Optimierungstechniken Dynamic Programming, Scanline Optimization, Graph Cuts und Simulated Annealing in Bezug auf den Parameter der Gewichtung der "optischen Glätte" (smoothness), der Werte zwischen 5 und 1000 annehmen kann und den Detailreichtum des Bildes kontrolliert. Als Matchingkosten werden unangehäufte absolute Differenzen benutzt.

3.7. Experiment 6

Verbesserung der Leistung des Graph Cut Algorithmus

In Experiment 6 wird der Graph Cut Algorithmus mit dem Ziel, die Ergebnisse aus Experiment 5 zu verbessern, getestet. Dafür wird bei mittlerer optischer Glätte das Verfahren von Birchfeld und Tomasi in Kombination mit anderen veränderten Parametern eingesetzt.

3.8. Subpixelverfeinerung

Alle Experimente wurden ohne Subpixelverfeinerung gemacht, um die Qualität der Verfahren bezüglich verschiedener Parameter unverfälscht beurteilen zu können.

Die Subpixelverfeinerung der Disparitäten, welche in jedem der Experimente 1 bis 6 anwendbar ist, wird von der boolean-Variablen `refine_subpix` kontrolliert, die in der Textdatei `param_in.txt` auf 1 gesetzt werden muss. Für die Berechnung werden die drei um die Gewinnerdisparität angehäuften Matchingkostenwerte untersucht. Durch drei Punkte kann eine Parabel eindeutig bestimmt werden [18]. In diesem Fall sind diese Punkte die Matchingkosten der Gewinnerdisparität sowie die der nächstgrößeren und nächstkleineren Disparität. Falls die Gewinnerdisparität d_{\min} ist, werden die Matchingkosten von d_{\min} und den beiden nächstgrößeren Disparitäten ausgewählt. Analog wird für d_{\max} als Gewinnerdisparität eine Parabel durch die Matchingkosten von d_{\max} und den Kosten der beiden nächstkleineren Disparitäten gelegt. Liegt das Minimum der Parabel innerhalb eines halben Disparitätsschrittes an der Gewinnerdisparität, so wird der Disparitätswert des Minimums als endgültige Disparitätsschätzung benutzt.

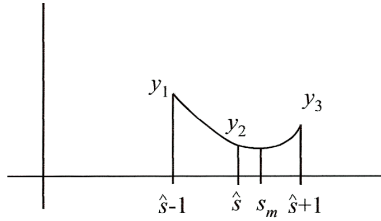


Abb. 12: Subpixelabschätzung [A4]

Seien die Disparitäten $s - 1$, s und $s + 1$ mit den Matchingkosten y_1 für $s - 1$, y_2 für s und y_3 für $s + 1$, so ergibt sich die Position der minimalen Matchingkosten aus dem Minimum der Parabel s_m , welches berechnet werden kann mit der Formel:

$$s_m = s + \frac{y_1 - y_3}{2(y_1 - 2y_2 + y_3)} \quad (8)$$

4. Berechnung der Disparitäten in selbst erzeugten Stereobildern

Nach dem Erzeugen des Programms kann mit Textdateien definiert werden, welches Experiment zum Erstellen der Tiefenkarten benutzt wird. In diesen Dateien werden auch die Namen der Input-Bilder festgelegt, sowie weitere Parameter, die teilweise zwingend angegeben werden müssen, teilweise zusätzlich hinzugefügt werden können.

Die Werte `disp_min`, `disp_max` und `disp_scale` sind für die Ausarbeitung festzulegen. Sie bezeichnen die minimale und die maximale Disparität im Bild sowie den Skalierungsfaktor für die Helligkeiten.

Die im Bild vorhandenen Disparitäten müssen also vorher grob abgeschätzt werden. Anhand von `disp_min` und `disp_max` kann `disp_scale` bestimmt werden: Die möglichen Helligkeitsabstufungen liegen zwischen 0 und 255 (0 = schwarz, 255 = weiß). Der Bereich zwischen der minimalen und der maximalen Disparität sollte optimal auf diese Werte verteilt werden.

Es gilt die Formel

$$H = (d - \text{disp_min}) \cdot \text{disp_scale} \quad (9)$$

wobei H = Helligkeit und d = Disparität.

$$\text{Somit gilt } H_{\text{max}} = (\text{disp_max} - \text{disp_min}) \cdot \text{disp_scale} \quad (10)$$

$$\text{und } \text{disp_scale} = \frac{H_{\text{max}}}{\text{disp_max} - \text{disp_min}} \quad (11)$$

Je größer die Disparitätsunterschiede sind, desto kleiner kann `disp_scale` gewählt werden.

4.1. Anfertigung von Stereobildern

Um das Programm zu testen, wurden verschiedenste eigene Motive hergestellt. Bei der Aufnahme muss darauf geachtet werden, dass die Objekte im Allgemeinen gut durchstrukturiert sind, da aus dem Versatz der Strukturen die Disparitäten ermittelt werden.

Zur Aufnahme wurde eine analoge Kleinbildkamera mit 135 mm Brennweite benutzt. Die Blende wurde soweit wie möglich geschlossen, um die Schärfentiefe so groß wie möglich zu machen.

Die Kamera wurde seitlich verschoben, die Ausrichtung bezüglich der Szene blieb unverändert, d.h. die optischen Achsen beider Aufnahmesituationen sind parallel. Dies

entspricht der Standard-Stereogeometrie, die in Abbildung 13a dargestellt ist. Ein Bildpunkt P wird im zweiten Bild um die Disparität d entlang der epipolaren Linie versetzt dargestellt.

Um eine exakte Kameraverschiebung zu gewährleisten, wurde auf dem Boden eine Schiene angebracht, an die ein Brett, auf dem sich die Kamera befand, angelegt wurde. Das Brett wurde nach jeder Aufnahme um 1 cm entlang der Schiene, auf der sich eine Skala befand, verschoben. Auf diese Weise entstanden Serien von Stereobildern, aus denen die Bilder, deren Disparitäten etwa in der richtigen Größenordnung waren, ausgewählt wurden.

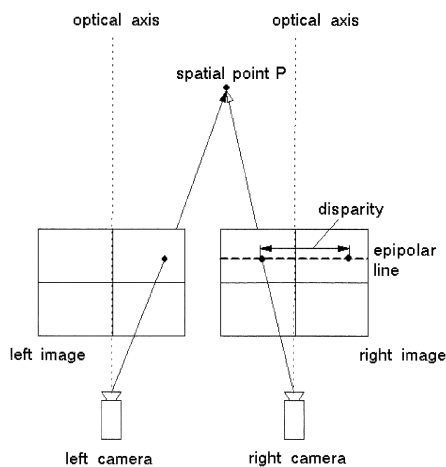


Abb. 13a: Standard-Stereogeometrie [A5]

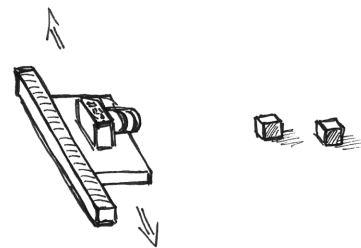


Abb. 13b: Aufbau zum Fotografieren von Stereobildern [A9]

Zum Testen der Stereoprogramme wurden Motive unterschiedlichster Art aufgenommen. Manche zeigen reelle Objekte wie Spielfiguren, Würfel, Glühbirnen etc. um die Leistung der Programme in der normalen menschlichen Umgebung zu testen. Andere Motive wurden speziell hergestellt, um die Qualität der Algorithmen unter verschiedenen Bedingungen zu erforschen. Dabei entstanden Bilder, die die Voraussetzungen für gute Ergebnisse unterstützen, aber auch Bilder, die absichtlich schwierige, z.B. texturlose Bildstellen enthalten, sowie Bilder, die dem Betrachter kaum Tiefeninformationen liefern. Zur Berechnung der Disparität muss die Kameraverschiebung bekannt sein. Der Abstand der einzelnen Objekte zur Kamera wurde ebenfalls gemessen. Die Negative wurden eindigitalisiert und so benannt, dass aus der jeweiligen Namensgebung die Aufnahmeposition hervorgeht. Die Bilder Kiwi08 und Kiwi10 unterscheiden sich dadurch, dass die Kamera zur Aufnahme von Kiwi10 um 2 cm nach rechts verschoben wurde.

Da die Verfahren sehr rechenintensiv sind und sich der Rechenaufwand bei großen Bildern extrem vervielfältigt, mussten die Bilder downsampled werden, d.h. die Zahl der Pixel

wurde verkleinert. Damit keine Aliasfehler auftreten, muss vorher eine Weichzeichnung durchgeführt werden. Bildgrößen von etwa 450 x 300 Pixel haben sich als zweckmäßig erwiesen. Auch die Urheber des Programms arbeiten mit Bildgrößen in diesem Bereich. Wie sich während der Experimente herausstellte, waren die Parameter auf diese Bildgrößen abgestimmt.

4.2. Berechnung der Helligkeit aus der Entfernung

Aus den entstandenen Bildern kann man an bestimmten Stellen die Helligkeitswerte ablesen und in Disparitäten umrechnen, das geschieht durch Umstellen der Formel (9) in

$$d = \text{disp_min} + \frac{H}{\text{disp_scale}} \quad (12)$$

Um Disparität und Entfernung ineinander umzurechnen, müssen einige Formeln ineinander eingesetzt werden.

Die Entfernung wurde zwischen Objekt und Filmebene gemessen. Die reine Gegenstandsweite, also die Strecke zwischen Objekt und objektseitiger Hauptebene des Objektivs ist nicht messbar, da die Lage der Hauptebenen nicht bekannt ist. Die gemessene Strecke, genannt a , setzt sich zusammen aus Gegenstandsweite g , Bildweite b und dem Abstand der Hauptebenen e . Es gelten die Formeln für zusammengesetzte Linsensysteme, also für "dicke Linsen".

$$a = g + e + b \quad (13)$$

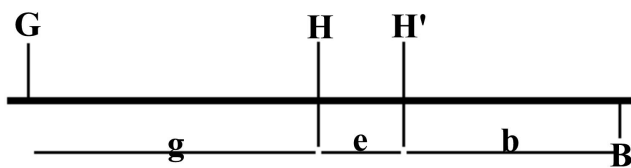


Abb. 14: Position von Hauptebenen, Bildweite und Gegenstandsweite [A9]

Dabei muss beachtet werden, dass bei der Aufnahme auf eine bestimmte Gegenstandsweite scharfgestellt wird, und Objekte, die sich vor oder hinter dieser Stelle befinden, eine andere Gegenstandsweite und somit einen anderen Abbildungsmaßstab haben. Das scharfe Bild solcher Objekte entsteht also nicht in der Filmebene. Da der Film jedoch das Bild aller Objekte einfängt, ist deren Bildweite gleich der Bildweite, die zu der Gegenstandsweite gehört, auf die scharfgestellt wurde. Da für Stereoaufnahmen eine Staffelung von Objekten im Raum unvermeidbar ist, muss die sonst entstehende Unschärfe durch das Schließen der Blende vermieden oder möglichst gering gehalten werden.

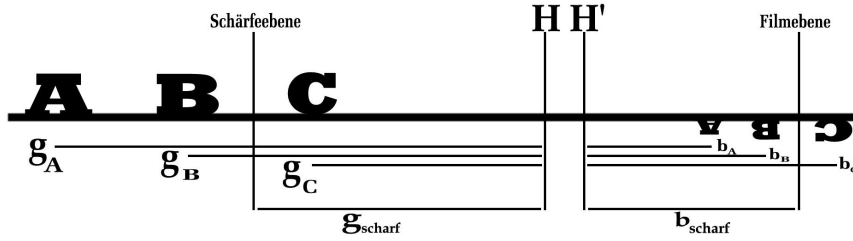


Abb. 15: Bildweiten und Gegenstandsweiten für in der Tiefe gestaffelte Objekte [A9]

Der Abbildungsmaßstab in der Schärfebene beträgt also

$$\beta_{neg} = \frac{b_{scharf}}{g_{scharf}} \quad (14)$$

der Abbildungsmaßstab an einer anderen Stelle x beträgt

$$\beta_{neg} = \frac{b_{scharf}}{g_x} \quad (15)$$

somit gilt auch für a:

$$a_x = g_x + e + b_{scharf} \quad (16)$$

Um den Abstand der Hauptebenen e für die scharfgestellte Position zu erhalten, werden die Formeln

$$g = f \left(1 + \frac{1}{\beta} \right) \quad (17)$$

$$b = f(1 + \beta) \quad (18)$$

in die Gleichung (13) eingesetzt:

$$\begin{aligned} a &= g + e + b \\ a &= f \left(1 + \frac{1}{\beta_{neg}} \right) + e + f(1 + \beta_{neg}) \\ a &= e + 2f + f \left(\frac{1}{\beta_{neg}} + \beta_{neg} \right) \\ e &= a - \left[2f + f \left(\frac{1}{\beta_{neg}} + \beta_{neg} \right) \right] \end{aligned} \quad (19)$$

Der benötigte Abbildungsmaßstab kann ermittelt werden, indem ein Testbild mit Skala, das sich exakt in der scharfeingestellten Position befindet, ab fotografiert wird.

$$\beta_{neg} = \frac{B}{G} \quad B = \text{Bildgröße}, G = \text{Objektgröße} \quad (20)$$

Die Bildgröße ist in diesem Fall die Größe des Bildes im Negativ.

Multiplizieren der Disparität $[cm]$ mit der Auflösung $r \left[\frac{Pixel}{cm} \right]$ ergibt die Disparität in Pixel.

$$d[Pixel] = d[cm] \cdot r \left[\frac{Pixel}{cm} \right] \quad (21)$$

Die Disparität $[cm]$ ergibt sich aus der Multiplikation der Kameraverschiebung p mit dem gesamten Abbildungsmaßstab β_{ges} .

$$d = p \cdot \beta_{ges} \quad (22)$$

Der gesamte Abbildungsmaßstab ist das Produkt der einzelnen Maßstäbe.

$$\beta_{ges} = \beta_{neg} \cdot \beta_{pos} \quad (23)$$

Der Maßstab β_{pos} ist wiederum $\frac{B}{G}$, wobei für B die Zahl der Pixel des Bildes in einer Richtung mit der Auflösung multipliziert wird, G ist in diesem Fall die entsprechende Negativgröße.

Der Maßstab β_{neg} kann innerhalb eines Bildes verschiedene Werte annehmen, es gilt Gleichung (15). Werden innerhalb eines Bildes die Disparitäten verschiedener Objekte berechnet, muss β_{neg} jeweils aus der Gegenstandsweite des Objekts und aus der Bildweite, die zu dem Objekt in der Schärfenebene gehört, berechnet werden.

Aus Gleichung (16) kann die jeweilige Gegenstandsweite ermittelt werden:

$$g_x = a_x - e - b_{scharf} \quad (24)$$

Setzt man die Gleichungen (15), (21), (22), (23) und (24) ineinander ein, ergibt sich für die Disparität:

$$d = p \cdot \beta_{pos} \cdot \frac{b_{scharf}}{a - e - b_{scharf}} \cdot r \quad (25)$$

Der daraus resultierende Helligkeitston ergibt sich durch Einsetzen in Gleichung (9).

Eine Funktion der Helligkeit in Abhängigkeit von der Entfernung für die Aufnahmesituation des Motivs Spiel mit $disp_{min} = 20$ und $disp_{scale} = 12$ wird in Abbildung 16 dargestellt.

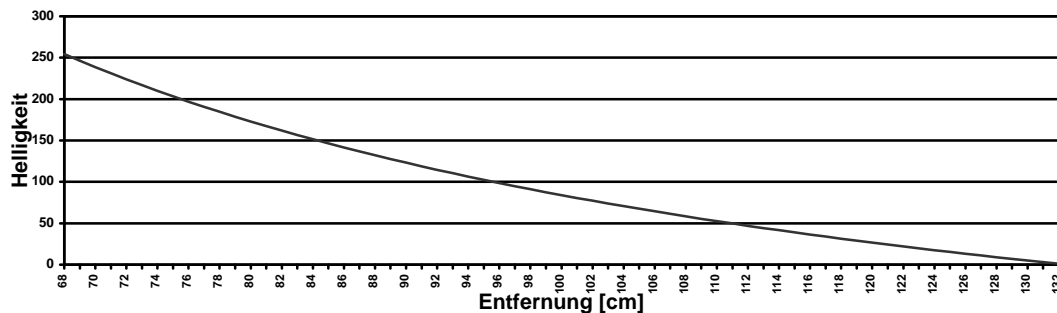


Abb. 16: Helligkeitstwerte der Tiefenkarte als Funktion der Entfernung der Objekte [A9]

Für die Berechnung in umgekehrter Richtung, also um die Entfernung a aus der Helligkeit zu erhalten, muss die nach a aufgelöste Gleichung (25) benutzt werden:

$$a = e + b_{\text{scharf}} + \frac{p \cdot \beta_{\text{pos}} \cdot r \cdot b_{\text{scharf}}}{d} \quad (26)$$

Dabei kann d aus der Helligkeit berechnet werden nach Formel (12).

4.3. Fehlerabschätzung

Um eine korrekte Helligkeitswiedergabe in den Tiefenkarten sicherzustellen, muss untersucht werden, ob störende Einflüsse, wie z.B. Abbildungsfehler des Objektivs oder verfälschte Entfernung eines Objektes durch Positionierung am Bildrand das Ergebnis beeinträchtigen.

Entsprechend Abbildung 17 ist die Entfernung g der Abstand eines in der Bildmitte liegenden Objekts A. Auf gleicher Höhe, jedoch am Bildrand, liegt Objekt B. Die Entfernung x ist jedoch aufgrund der der Fotografie zugrunde liegenden Zentralprojektion größer als g , was im Bild einen kleineren Abbildungsmaßstab zur Folge haben könnte. Daraus würden kleinere Disparitäten berechnet werden, die in der Tiefenkarte dunkler dargestellt werden würden und eine zu große Entfernung vortäuschen würden.

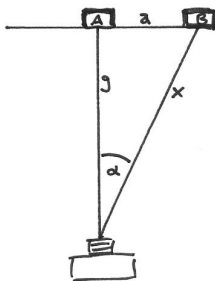


Abb. 17: Entfernungen von Objekten in der Bildmitte (A) und am Bildrand (B) [A9]

g lässt sich errechnen durch Multiplikation von x mit $\cos \alpha$, wobei $\alpha = \arctan \frac{a}{g}$.

$$g = x \cdot \cos \left(\arctan \frac{a}{g} \right) \quad (27)$$

In der konkreten Aufnahmesituation beträgt die halbe Breite des Originalaufbaus a etwa 6,8 cm, g beträgt 66,86 cm. Durch Umstellen der Formel (27) ergibt sich für x eine Entfernung von 67,20 cm. Die Differenz beträgt 0,34 cm, dies entspricht einem Fehler von 0,5 %, der vernachlässigt werden kann.

In einer Aufnahmesituation mit Normalbrennweite wäre der Fehler mit etwa 5 % abzuschätzen.

Da ein Objektiv von Natur aus mit Abbildungsfehlern behaftet ist, wurden Testbilder aufgenommen, um die Art und Größe der Fehler abzuschätzen. Die Testbilder lassen eine kissenförmige Verzeichnung erkennen, die in den Tiefenkarten zu einer verfälschten Darstellung führt. Durch die Verzeichnung wird die Bildmitte kleiner als die Bildränder wiedergegeben, was dazu führt, dass die Disparitäten in der Mitte verhältnismäßig kleiner als am Rand erscheinen und entsprechend in der Mitte dunkler und am Rand heller dargestellt werden.

Um diesen Fehler zu beheben, musste ein Programm zur Entzerrung der Bilder geschrieben werden, als Plugin für das auf Java basierende Bildbearbeitungsprogramm ImageJ. Das Programm ist Freeware und ist erhältlich auf der Internetseite [16]. Ein Tutorial zum Schreiben von ImageJ-Plugins ist erhältlich unter [17].

In einem Testbild werden Referenzpunkte ausgewählt und deren Ist- und Sollwerte bestimmt. Da das Testbild ein kariertes Blatt Papier zeigt, wurde ein gleiches Blatt eingescannt und halbtransparent über das Bild gelegt. Auf diese Weise können die jeweiligen x- und y- Werte gewonnen und in eine Look Up Table geschrieben werden. Im Programmablauf wird die Position eines jeden Soll-Bildpunktes im entzerrten Bild bezüglich der ihn umgebenden Soll-Referenzpunkte bestimmt und die entsprechende Position im verzerrten Bild anhand der Ist-Referenzpunkte berechnet. Da die berechnete Position zwischen vier Pixeln liegen kann, muss der Farbton aus den Farbwerten der umliegenden Pixel des verzerrten Bildes interpoliert werden und in das entzerrte Bild eingetragen werden. Einzelheiten über die Entstehung des Programms können dem Anhang entnommen werden.

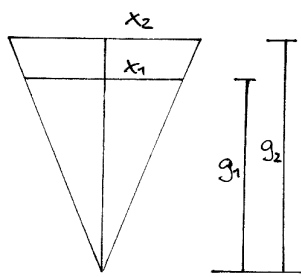
Der Abbildungsmaßstab β_{pos} wird durch die Entzerrung verändert und muss neu berechnet werden.

Da die Größen, mit denen gerechnet wurde, aufgrund von Messungenauigkeiten fehlerbehaftet sind, wird es nötig, die Größe des maximalen Fehlers zu bestimmen. Zunächst wird der Fehler jeder einzelnen Größe bestimmt, dann die Fehler zusammengerechnet. Laut [14] ist bei einer Summe oder Differenz von Messgrößen der *absolute* Maximalfehler gleich der Summe der Beträge ihrer Absolutfehler. Bei einem Produkt oder Quotienten von Messgrößen ist der *relative* Maximalfehler gleich der Summe der Beträge ihrer Relativfehler.

Beispielsweise ergibt sich für die dargestellte Helligkeit der absolute Fehler einer halben Helligkeitsstufe, also $0,5 * \text{disp_scale}$. Dieser Fehler kann durch Subpixelverfeinerung oder einen Wert für disp_step kleiner als 1 verringert werden.

Nach der schrittweisen Berechnung mit den oben genannten Verfahren ergibt sich ein relativer Maximalfehler von 16 %. Dabei wurde die auf dem Objektiv angegebene Brennweite von 135 mm nur mit einem absoluten Rundungsfehler von 0,5 mm belastet. Glaubt man jedoch den Berichten von Stiftung Warentest [15], so darf die tatsächliche Brennweite eines Objektivs um bis zu 5 % vom angegebenen Wert abweichen, manche Hersteller liegen über 5 %. Rechnet man diese Ungenauigkeit ein, kommt man in der gegebenen Aufnahmesituation auf einen Relativfehler von etwa 44 %. Somit wird eine exakte Berechnung erschwert, und für professionelle Anwendungen muss der tatsächliche Wert der Brennweite des Objektivs kontrolliert werden.

Eine Möglichkeit, die wirkliche Brennweite zu erhalten, ist die experimentelle Ermittlung der Gegenstandsweite, aus der alle weiteren unbekannten Größen hergeleitet werden können. Dazu wird die Kamera entsprechend der Aufnahmesituation der Stereobilder scharfgestellt und in die fokussierte Ebene sowie in einem definierten Abstand hinter dieser Ebene je eine horizontale Skala mit Zentimetereinteilung angebracht. Für Werte auf der vorderen Skala am Bildrand lassen sich Vergleichswerte auf der hinteren Skala ablesen. Mit den nun bekannten Werten x_1 , x_2 und $(g_2 - g_1)$ kann gemäß den Eigenschaften des Strahlensatzes die gesuchte Gegenstandsweite g_1 berechnet werden:



$$\frac{g_2 - g_1}{x_2 - x_1} = \frac{g_1}{x_1} \quad (28)$$

Abb. 18: Strahlensatz [A9]

Auflösen nach g_1 ergibt

$$g_1 = x_1 \cdot \left(\frac{g_2 - g_1}{x_2 - x_1} \right) \quad (29)$$

Da bei der optischen Abbildung der Mittelpunktstrahl die Linse ungebrochen durchläuft und durch die Hauptebenen nur parallel verschoben wird, ist der Bildwinkel zwischen Mittelpunktstrahl und optischer Achse auf der Objektseite und der Bildseite gleich groß.

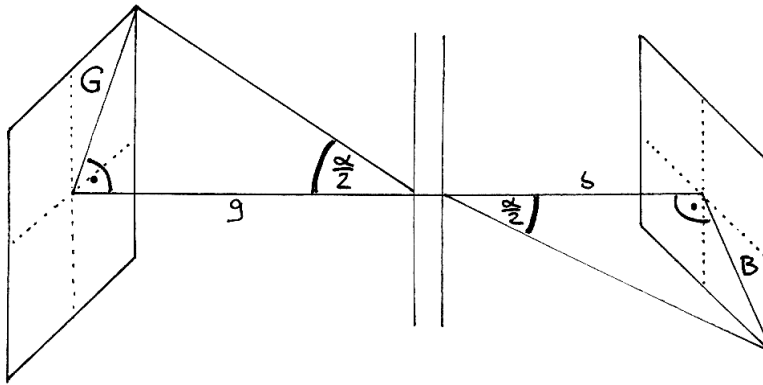


Abb. 19: Bildwinkel [A9]

Daher gilt die Formel

$$\tan \frac{\alpha}{2} = \frac{G}{g} = \frac{B}{b} \quad (30)$$

α stellt den gesamten Bildwinkel dar.

G und B sind die halbierten Diagonalen in Original und Bild, deren Werte ermittelbar sind. Somit können die Bildweite b und alle anderen Größen berechnet werden. Im konkreten Fall ergab sich eine leichte Abweichung der Brennweite, deren Auswirkung auf die Berechnungen der Entfernungen vernachlässigt werden kann.

Da alle Bilder unter den gleichen Aufnahmebedingungen entstanden und sich die Fehler zum Teil gegenseitig kompensieren, kann in Wirklichkeit eine größere Genauigkeit angenommen werden. Vernachlässigt man den gleich bleibenden Fehler während der Aufnahme einer Serie von Bildern eines Motivs, beträgt der Fehler nur noch 6 %. Davon ist in der konkreten Situation die größte Fehlerquelle die Kameraverschiebung, die jeweils nur 1 cm beträgt. Da sich Stereobilder mit diesem Positionsunterschied am besten eignen, ist die Fehleranfälligkeit sehr groß. Wird bei jedem Bild eine Ungenauigkeit eines viertel Millimeters einkalkuliert, liegt der Fehler schon bei 5 %. Aus gestalterischen Gründen sowie zur Vermeidung anderer Fehlerquellen, wie verfälschte Abbildung eines Objektes am Bildrand, die oben beschrieben wurde, fiel die Entscheidung zugunsten dieses Aufbaus.

4.4. Vergleich des gemessenen mit dem berechneten Abstand

Um die tatsächliche Genauigkeit der Berechnungen in Erfahrung zu bringen, sollen nun die Helligkeitswerte der Motive aus den Tiefenkarten für eine Berechnung des Abstandes in cm herangezogen werden und mit den während der Aufnahme gemessenen Entfernungen verglichen werden. Dafür wird die oben hergeleitete Formel (26) benutzt. Als Beispiel

dient das Motiv Spiel. Die Bildgröße beträgt 450 x 300 Pixel, $\text{disp_min} = 20$, $\text{disp_scale} = 12$. Die Tiefenkarte wurde mit dem Graph Cut Verfahren erstellt.

Helligkeit	Disparitaet	Entfernung	gemessene Entfernung	Motiv
228.0	39.0	71.43308734861537	70	Figur Blau
240.0	40.0	69.8256601649	69	Figur Lila
204.0	37.0	74.90860558367567	71	Figur Rot
180.0	35.0	78.78132590274285	75	Figur Orange
168.0	34.0	80.88854137047058	77	Figur Gelb
132.0	31.0	88.02588408374193	86	Figur Grau
96.0	28.0	96.69265737842856		Hintergrund
84.0	27.0	100.00957061466666	95	Hintergrund
72.0	26.0	103.58163102292308		Hintergrund

Der relative Fehler liegt hier bei maximal 5 %. Alle Tabellen mit den gemessenen und errechneten Werten der einzelnen Motive befinden sich im Anhang A.2. Bei einigen Motiven stimmen die Werte relativ gut überein, bei einigen anderen gibt es systematische Abweichungen, das heißt, alle Entfernungen sind beispielsweise um einen bestimmten Faktor zu klein. Vermutlich liegt die Ursache der Abweichungen in der in Kapitel 4.3. besprochenen Tatsache, dass die Kameraverschiebung von nur 1 cm ausgesprochen fehleranfällig ist. Jedoch ist es trotz der Ungenauigkeiten möglich, die Werte für die Entfernungen verschiedener Bildstellen ungefähr zu rekonstruieren. Somit sind die Voraussetzungen für eine dreidimensionale Wiedergabe einer Originalszene gegeben.

4.5. Erstellung von Groundtruth-Karten

Die Erstellung einer Groundtruth-Karte ist für eine quantitative Fehlerschätzung unerlässlich. Jedoch besteht die Problematik, dass eine ideale Tiefenkarte entworfen werden muss, basierend auf vorgegebenen fehlerbehafteten Werten. Eine Methode für die Zuordnung der Grauwerte wäre, die bei der Aufnahme mit dem Zollstock gemessenen Entfernungen in Disparitäten und dann in Helligkeiten umzurechnen. Dabei könnte der oben erwähnte maximale Fehler von etwa 16 % entstehen. Vergleicht man die Helligkeitswerte der jeweils besten Tiefenkarten aus Serien, die mit verschiedenen Verfahren erstellt wurden, fällt auf, dass zwar die Art der Darstellung unterschiedlich ist, aber die Helligkeitswerte sehr gut übereinstimmen, so dass jedem Bildteil ein konkreter Wert zugeordnet werden kann. Um die Disparitäten eines Motivs visuell zu überprüfen, werden die Stereobilder um den Betrag einer bestimmten Disparität gegeneinander verschoben. Ein Differenzbild beider Bilder lässt die Stellen mit der entsprechenden Disparität schwarz erscheinen. Ein solches Bild, also der x-y-Schnitt eines DSI, kann mit

dem Programm erstellt werden, indem die Variable `verbose` für die Ausführlichkeit der Berichterstattung erhöht wird. Abbildung 4 in Kapitel 2.2. zeigt solche DSI-Schnitte.

Zur Erstellung der Groundtruth-Karte wird das Referenzbild eines Motivs einfach mit den entsprechenden Grauwerten "angemalt". Die Groundtruth-Karte ist an vorher festzulegende Werte von `disp_min`, `disp_max` und `disp_scale` gebunden, die auch zur Erstellung der Tiefenkarten gelten. Sind für eine Groundtruth-Karte diese Werte nicht bekannt, oder stimmen sie nicht mit denen der Tiefenkarten überein, kann keine Fehlerauswertung erfolgen.

Die Variable `eval_bad_tresh` bestimmt die maximal zulässige Abweichung des Disparitätswertes eines Pixels. Sie hat in allen Experimenten den Wert 1, kann jedoch nach Belieben verändert werden.

Die Art der zu bewertenden Stereobilder prägt das Aussehen der Groundtruth-Karte. In einigen Karten sind nur zwei oder drei Grauwerte vorhanden, in anderen deutlich mehr. Die Größe des Fehlers ist dabei nicht immer gleichzusetzen mit der Aussagekraft der Tiefenkarte. Besteht die Groundtruth-Karte beispielsweise nur aus zwei Grauwerten, wie im Motiv "Chaos_flach", stellt das Graph Cut Verfahren bei falscher Wahl der Parameter das gesamte Bild in dem dunkleren Ton dar. Das Bild ist unbrauchbar, der Fehler liegt aber nur bei 50 %. Im Vergleich dazu erreichen Tiefenkarten komplizierter Motive bis zu 60 % Fehlerquoten, können jedoch für Tiefeneinschätzungen verwendet werden. Das RMS-Fehlermaß sowie der prozentuale Fehler sollten daher nur im Bezug auf ein bestimmtes Stereobildpaar zur Bewertung herangezogen werden.

5. Tests und Ergebnisse

5.1. Ermittlung der Leistung der verschiedenen Verfahren und Vergleich mit den Angaben der Autoren

Mit den prozentualen Fehlern aus den Tiefenkarten der Stereobilder soll die Qualität der Verfahren gemessen und eine Art Rangordnung erstellt werden. Diese Aufstellung kann nun mit der Einschätzung der Autoren Scharstein und Szeliski verglichen werden, da dieselben Experimente benutzt werden, und somit dieselben Rahmenbedingungen geschaffen werden. In der Auswertung werden alle Pixel bzw. alle nicht verborgenen Pixel betrachtet. Fehlerstatistiken für Diskontinuitäten und texturlose Stellen wurden nicht ausgewertet, da sie sich nur auf einen kleinen Teil der im Bild vorkommenden Pixel beziehen und der Schwerpunkt der Arbeit auf den Vergleich der Gesamtleistungen der Algorithmen gelegt wird. Scharstein und Szeliski veröffentlichten in ihrem Bericht [1] Diagramme, die Fehlerauswertungen für Bereiche ohne Struktur und mit Tiefendiskontinuitäten enthalten.

Die Verfahren Graph Cuts, SSD (Sum of Squared Differences, lokale Methode), Scanline Optimization und Dynamic Programming liefern für jedes der Testbilder Tsukuba, Sawtooth, Venus und Map einen prozentualen Fehlerwert. Die lokale Methode SSD wird hier mit einem Minfilter benutzt und wurde stellvertretend für alle lokalen Methoden ausgewählt, da sie in den Tests der Autoren die besten Ergebnisse lieferte. Die folgende Tabelle zeigt die prozentualen Fehlerwerte für optimierte Parameter. Die Werte stammen aus [1], S.31. Der Wert für Bayesian Diffusion kann nicht verglichen werden, da das Verfahren nicht implementiert ist. Stattdessen wären Werte für die erweiterten lokalen Methoden (Minfilter, Diffusion, Membrandiffusion) sowie Simulated Annealing wünschenswert gewesen, sie werden jedoch nicht erwähnt.

	Tsukuba	Sawtooth	Venus	Map	Durchschnitt
GC	1,94	0,98	1,48	0,09	1,13
SSD	5,23	1,55	2,92	0,22	2,48
SO	4,66	3,47	8,31	1,04	4,37
DP	3,82	3,70	9,13	1,21	4,47

Prozentuale Fehlerwerte verschiedener Algorithmen für die Stereobilder von Scharstein und Szeliski

Das Graph Cut Verfahren schneidet am besten ab, gefolgt von der lokalen Methode SSD. Etwas schlechtere Leistungen bringen Scanline Optimization und Dynamic Programming. Als Referenz wurden Ergebnisse mit optimierten Parametern herangezogen, da standardisierte Parameter das Ergebnis verfälschen könnten. Die Wirkung eines solchen Parameters ist sehr stark von Bildgröße und maximaler Disparität abhängig. Daher scheint die "fairste" Lösung der Vergleich der jeweils besten Ergebnisse einer Serie zu sein.

Die gleiche Berechnung soll nun mit den selbst angefertigten Stereobildern erfolgen. Dabei muss beachtet werden, dass die Fehler in einer anderen Größenordnung liegen werden, da die meisten Bilder von größerer Komplexität sind und an die selbst erstellten Groundtruth-Karten kein Anspruch an Perfektion gestellt werden kann.

Aus jeder Serie eines Algorithmus wird das Bild mit dem jeweils geringsten Fehler ausgewählt. Dabei findet grundsätzlich keine Subpixelverfeinerung statt und der Disparitätsschritt ist 1. Fünf verschiedene Fehlerwerte wurden erfasst: Der Prozentsatz der falsch gematchten Pixel im gesamten Bild, der Prozentsatz der falsch gematchten Pixel in nicht verborgenen Gebieten, der RMS-Fehler aller Pixel, der RMS-Fehler in nicht verborgenen Gebieten und der Prediction Error.

Die Fehlerwerte aller 16 Motive werden gemittelt und ergeben für jedes Experiment und jedes Verfahren einen Durchschnittswert. Alle Werte sind im Anhang in Tabellen zu finden. Für den Vergleich mit den Ergebnissen der Autoren Scharstein und Szeliski wurde jeweils das beste Ergebnis von Graph Cuts, Dynamic Programming, Scanline Optimization und das beste Ergebnis der lokalen Methoden, WTA benutzt. Fehlermaß ist der Prozentsatz der falsch zugeordneten nicht verborgenen Pixel.

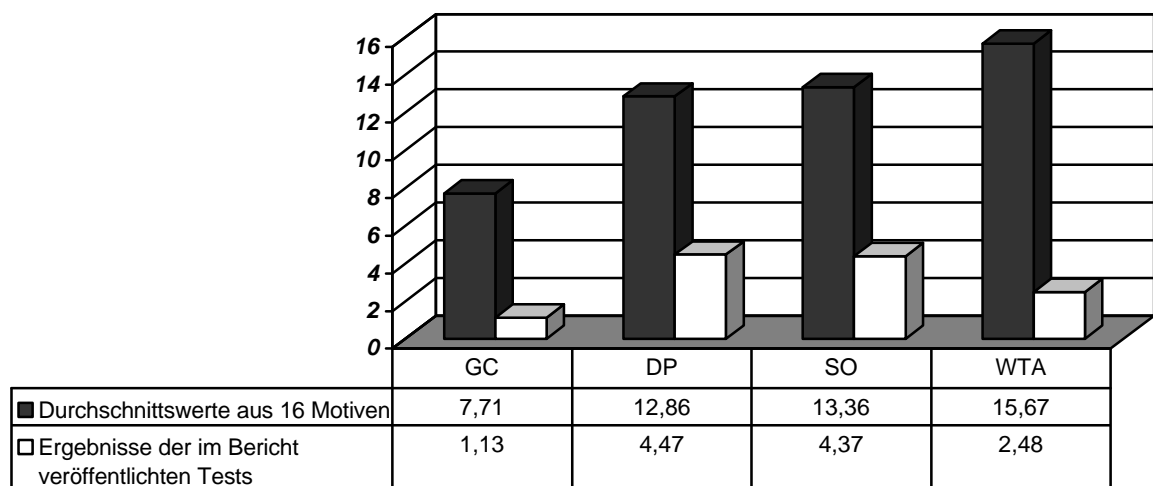


Abb. 20: Vergleich der Ergebnisse aus dem Bericht von Scharstein und Szeliski mit den eigenen Ergebnissen [A9]

In beiden Tests schneidet das Graph Cut Verfahren am besten ab. Dynamic Programming und Scanline Optimization liefern jeweils etwa gleich gute Ergebnisse, während die Leistung der lokalen Methoden unterschiedlich ist. Wie erwartet, liegt die Fehlerrate der selbst erstellten Motive deutlich über der der Testbilder der Autoren.

Da die relativen Leistungen von Graph Cuts, Dynamic Programming und Scanline Optimization annähernd gleich bewertet wurden, stellt sich die Frage, warum die lokalen Methoden in den Tests der Autoren besonders gut abschneiden. Vergleicht man die prozentualen Fehlerwerte der einzelnen Motive, so fällt auf, dass die lokale Methode SSD nur für die Bilder Sawtooth, Venus und Map jeweils das zweitbeste Ergebnis liefert, jedoch für Tsukuba das schlechteste. Diese Bilder befinden sich auf einer der beiliegenden CDs. Beim Betrachten wird klar, dass die Motive Sawtooth, Venus und Map jeweils nur aus einfachen strukturierten Flächen bestehen. Tsukuba zeigt jedoch eine realitätsnahe Situation eines Zimmers mit verschiedenen Gegenständen. Für dieses Motiv müssen die Leistungen der lokalen Methoden und auch von Scanline Optimization hinter Graph Cuts und Dynamic Programming eingestuft werden. Ein ähnliches Ergebnis liefert die Fehlerauswertung der 16 selbst erstellten Motive. Offensichtlich können die lokalen Methoden in einfachen Bildern gute Leistungen bringen, lassen jedoch in schwierigen Situationen deutlich nach.

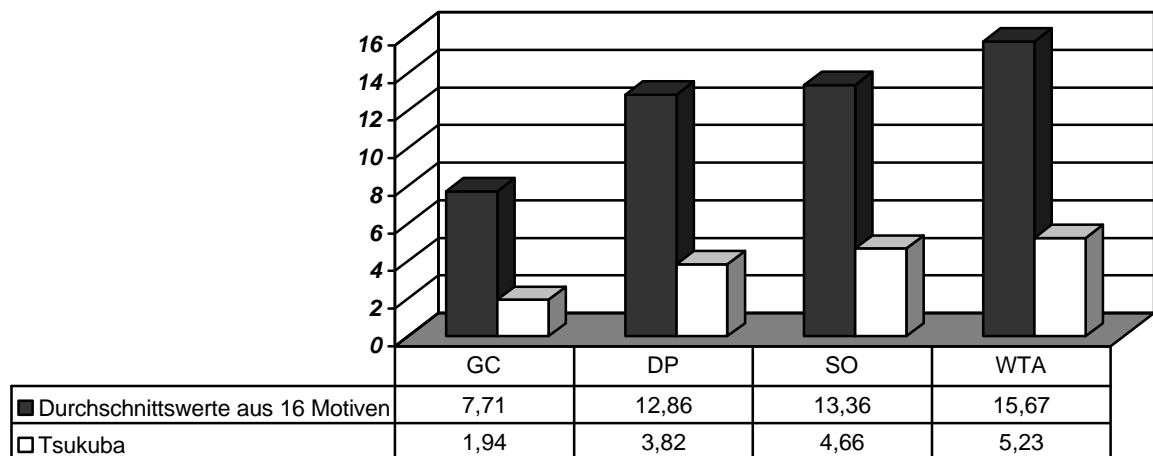


Abb. 21: Vergleich der Ergebnisse des Motivs Tsukuba mit den Durchschnittswerten der 16 eigenen Motive [A9]

Die Ergebnisse aller Verfahren aus allen Experimenten können dem folgenden Diagramm entnommen werden. Die Verfahren sind von links nach rechts entsprechend ihrer Leistung angeordnet.

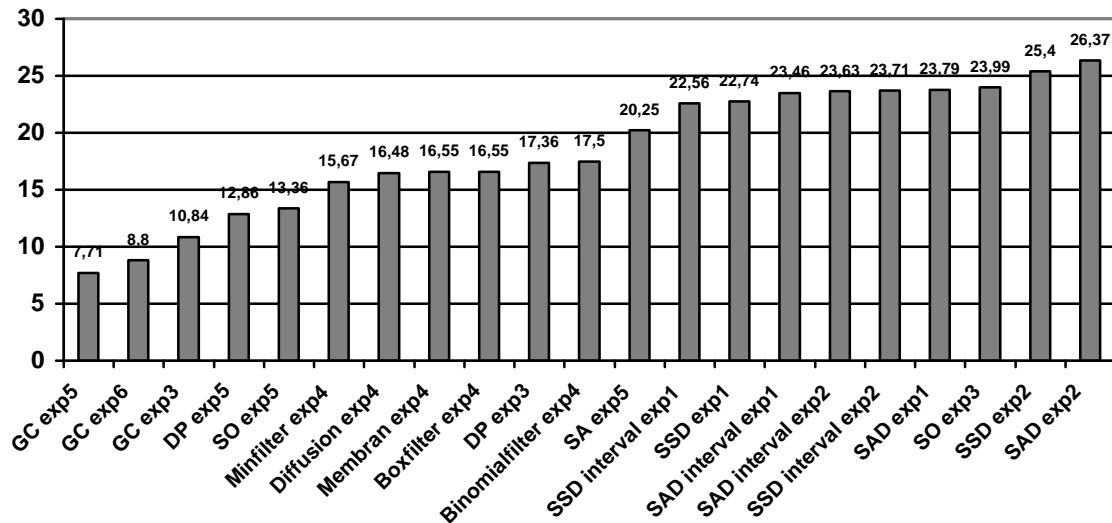


Abb. 22: Prozentsatz der Fehlerwerte nicht verborgener Pixel, Durchschnitt aus 16 Motiven [A9]

Beste Ergebnisse leistet das Graph Cut Verfahren, jedoch schneidet die Versuchsreihe aus Experiment 5 besser ab als die aus Experiment 6, welches eigentlich die Leistung optimieren sollte. Wie oben erwähnt, zeigen Dynamic Programming und Scanline Optimization ebenfalls gute Leistungen, gefolgt von der besten lokalen Methode, SAD mit Minifilter, sowie weiteren lokalen Methoden aus Experiment 4.

Nur mäßige Ergebnisse liefern die Experimente 1 und 2, wobei beachtet werden muss, dass jedes der Experimente wie oben beschrieben einen bestimmten Zweck verfolgt. Die Experimente 1 und 2 testen die Matchingkostenberechnung unter Variation der entsprechenden Parameter, variieren jedoch nicht die Fenstergröße oder die Art der Anhäufung. Somit können die Ergebnisse aus Experiment 1 und 2 nicht mit den anderen Ergebnissen gleichgesetzt werden, jedoch können sie untereinander verglichen werden. Offensichtlich liegen alle Fehlerwerte dieser beiden Experimente in der gleichen Größenordnung, wobei Experiment 1 (Boxfilter) Experiment 2 (Minifilter) leicht überlegen ist. Das Verfahren von Birchfeld und Tomasi (interval) verbessert das Ergebnis nur um etwa 1 %.

5.2. Vergleich des RMS-Fehlers aus Tiefenkarte und Groundtruth-Karte mit dem RMS-Fehler des Prediction Error

Wie oben erläutert, misst der RMS-Wert des Prediction Error den Unterschied zwischen einem aus einem Stereobild mit einer Tiefenkarte vorhergesagten Bild und dem entsprechenden Originalbild. Es soll nun überprüft werden, ob die Fehlermessung mit dem

Prediction Error die Verfahren in einer ähnlichen Reihenfolge bewertet wie die Standardbewertung. Das folgende Diagramm zeigt die Ergebnisse für alle fünf Fehlermaße.

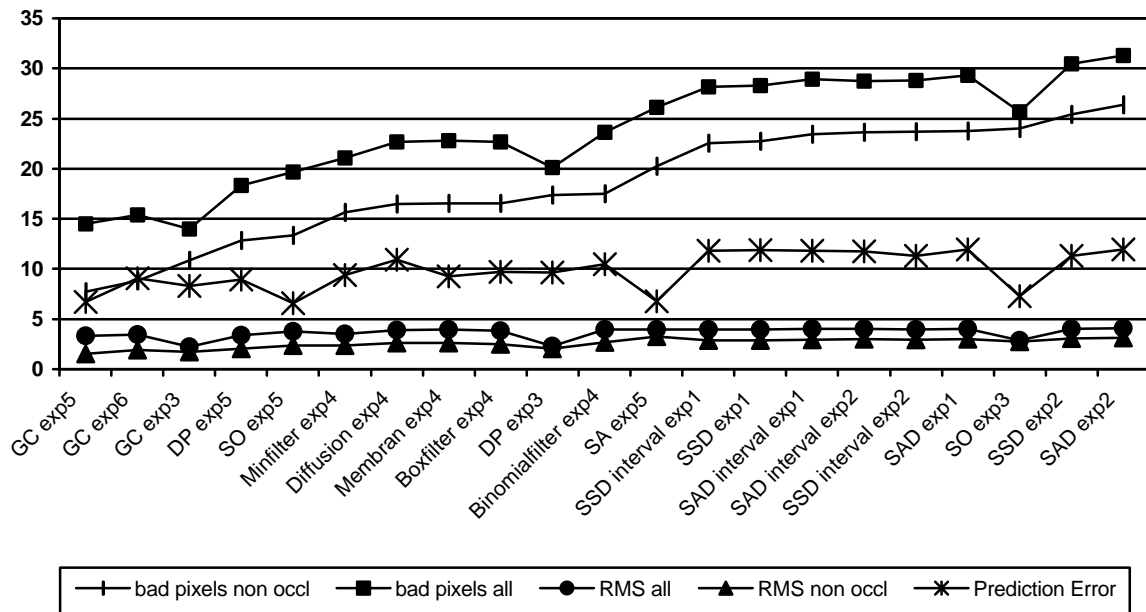


Abb. 23: Bewertung der Stereoverfahren mit verschiedenen Fehlermaßen [A9]

Die Skalierung der Y-Achse entspricht Prozentwerten für die falsch gematchten Pixel, beziehungsweise Disparitätseinheiten für die RMS-Fehler und den Prediction Error. Die Verfahren sind in der Reihenfolge des Prozentsatzes der schlechten nicht verborgenen Pixel (bad pixels non occluded) angeordnet. Es fällt auf, dass die Werte des Prediction Error kaum mit den Werten anderer Fehlermaße korrespondieren. Ordnet man die Verfahren entsprechend ihrem Prediction Error ein, so ergibt sich folgende Rangordnung:

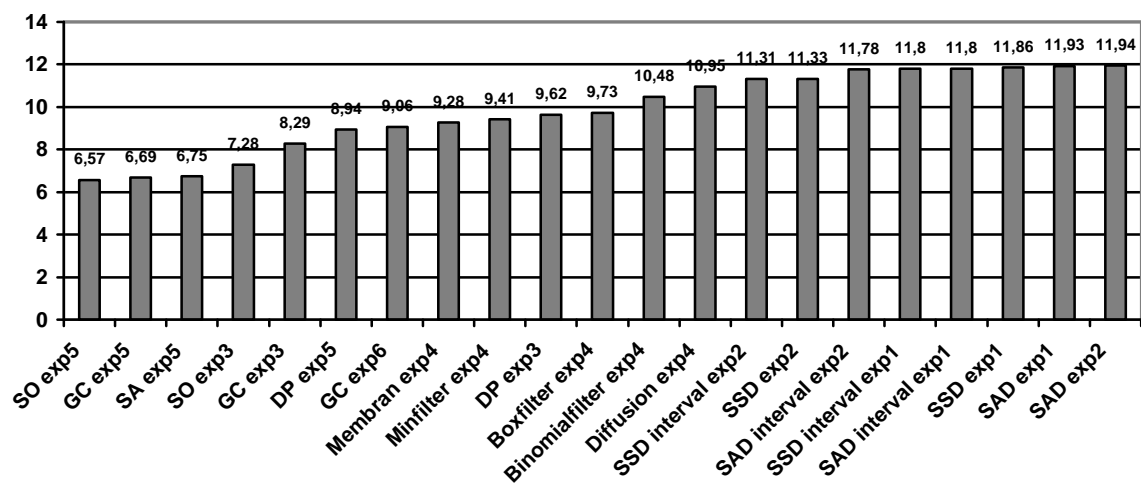


Abb. 24: Anordnung der Verfahren nach den Werten des Prediction Error [A9]

Scanline Optimization hat den niedrigsten Prediction Error, gefolgt von Graph Cuts und Simulated Annealing. Unter den lokalen Methoden schneidet die Membrandiffusion am besten ab. Die Ergebnisse des Prediction Error unterscheiden sich sehr stark von allen anderen Ergebnissen. Alle Werte, aus denen die Diagramme dieses Kapitels entstanden sind, stehen in der Tabelle A.4, "Endergebnis der Fehlerauswertung".

Betrachtet man einzelne Messreihen verschiedener Motive, so erkennt man weitere Unstimmigkeiten. In Experiment 1 und 2 verhält sich der Prediction Error ähnlich wie alle anderen Fehlermaße. Dies zeigt sich in der Statistik A.5, die die Zahl der "Volltreffer", d.h. der besten Ergebnisse einer bestimmten Parameterkombination einer Serie dokumentiert. Alle weiteren Experimente zeigen für den Prediction Error eine nahezu gegenläufige Tendenz im Vergleich zu anderen Fehlerwerten. Bilder mit ausgesprochen großem prozentualen Fehler erhalten den besten Prediction Error. Als Beispiel dient die Serie Minfilter des Experiments 4 für das Motiv Spiel. Prediction Error und RMS-Fehler sind abhängig von dem Parameter Fenstergröße aufgetragen. Die Ungleichheit der Bewertung ist nicht zu übersehen.

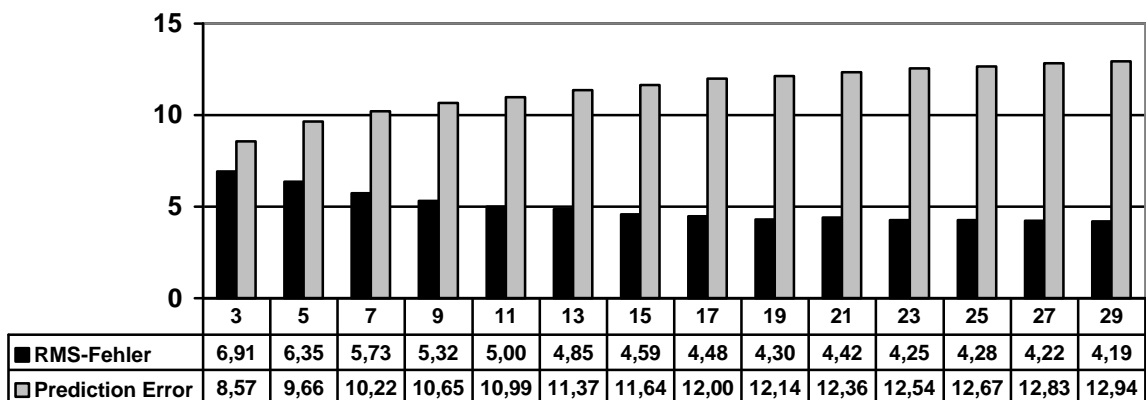


Abb. 25: RMS-Fehler und Prediction Error für Experiment 4 mit Minfilter für Motiv Spiel in Abhängigkeit von der Fenstergröße [A9]

Betrachtet man die Verfahren Diffusion und Membrandiffusion im Hinblick auf prozentuale Fehlermaße, so stellt man bei den meisten der 16 Motive fest, dass das Diffusionsverfahren mit steigender Anzahl der Iterationen immer bessere Werte ergibt. Die maximale Iterationszahl von 150 gibt im Allgemeinen Bestleistungen.

Das Membrandiffusionsverfahren ist, wie oben erwähnt, eine Weiterentwicklung der regulären Diffusion mit dem Kontrollwert β . Ist der Wert 0, handelt es sich im Prinzip um reguläre Diffusion. In den Messreihen werden Werte von 0,9 bis 0 benutzt. Bei nahezu allen Motiven liefert der β -Wert 0 die geringsten prozentualen Fehler. Somit liefern reguläre Diffusion und Membrandiffusion gleiche Ergebnisse und der Zweck der Membrandiffusion wird in Frage gestellt. Betrachtet man jedoch die Bewertung mit dem

Prediction Error, so erhält man für den β -Wert 0 schlechte Werte und für größeres β entsprechend bessere. Die Leistung der Membrandiffusion kommt erst mit Hilfe des Prediction Error zur Geltung.

5.3. Testen der Leistung der Algorithmen an texturlosen Stellen

Dieser Test stellt eine Art Härte-Test für die Verfahren dar. Das Motiv Papier besteht zu mehr als 50 % aus texturlosen Stellen. Der gesamte Vordergrund besteht aus unstrukturiertem weißem Papier. Die Leistung der Verfahren in diesem Test soll mit der normalen Leistung verglichen werden, um beurteilen zu können, welche Qualität die Verfahren unter schwierigen Bedingungen liefern.

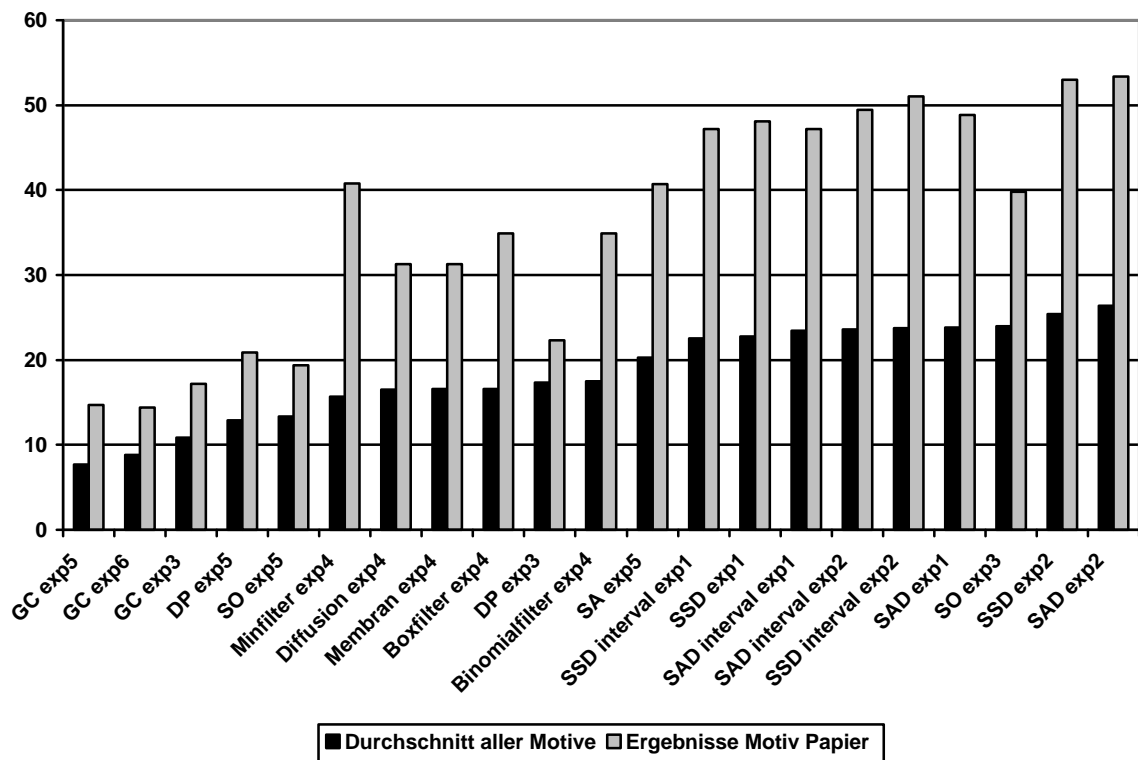


Abb. 26: Prozentsatz der Fehlerwerte von nicht verborgenen Pixeln:

Vergleich der Werte des Motivs Papier mit den Durchschnittswerten [A9]

In das Diagramm der durchschnittlichen Leistung aus 16 Motiven wurde die Leistung der Verfahren für das Motiv Papier eingefügt. Hier zeigt sich die hohe Qualität der globalen Methoden (Experimente 3, 5, 6).

Während die lokalen Methoden (Experimente 1, 2, 4) nur die Ränder des texturlosen Objekts in der Tiefenkarte richtig darstellen können und die restliche Fläche nur Rauschen enthält, was zu großen prozentualen Fehlerwerten führt (Abb. 27a), erfassen vor allem

Dynamic Programming und Graph Cuts das gesamte Objekt und füllen es mit dem richtigen Helligkeitswert (Abb. 27b).

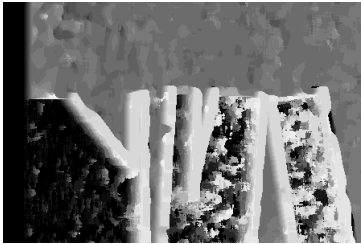


Abb. 27a: Tiefenkarte, SAD [A9]



Abb. 27b: Tiefenkarte, GC [A9]

Interessant ist die Beobachtung, dass auch die globalen Verfahren nicht in der Lage sind, der Bildstelle links unten einen Helligkeitswert zuzuordnen. Die Ursache dafür liegt wohl in der fehlenden seitlichen Begrenzung des Objekts.

5.4. Optimierung der Parameter

Es fällt auf, dass die Wahl der Parameter in den Standardscripthen manchmal lückenhaft ist und scheinbar nicht die beste Leistung eines Verfahrens offenbart. Da in jedem Experiment im Allgemeinen nur ein Parameter variiert wird, während alle anderen konstant gehalten werden, wird auf das Erzielen von optimalen Ergebnissen zugunsten der Erforschung der Wirkung der einzelnen Parameter verzichtet. Durch das Setzen von anderen Werten und der Kombination verschiedener Parameter soll nun versucht werden, die Ergebnisse zu optimieren.

Als Beispiel dient Simulated Annealing aus Experiment 5. Mit größerer optischer Glätte werden die mosaikähnlichen Flecken der Tiefenkarte immer größer, jedoch sind bei maximaler optischer Glätte noch keine durchgehenden Flächen entstanden. Aus jedem Algorithmus soll nun experimentell das Optimum an Leistung herausgeholt werden.

5.4.1. Optimierung der Experimente 1 und 2

Die Experimente 1 und 2 wurden entsprechend der Vorgaben in den Textdateien bei konstanter Fenstergröße 9 durchgeführt, um die Anhäufung mit absoluten und quadrierten Differenzen zu vergleichen, die Wirkung des Birchfeld-Tomasi-Verfahrens sowie die Unterschiede von Boxfilter und Minfilter zu erforschen.

Aus Experiment 4 lässt sich jedoch ablesen, dass die optimale Fenstergröße für die selbst erstellten Bilder in einem anderen Bereich liegt, nämlich etwa zwischen 23 und 29, siehe

Tabelle A.5. Da zwischen den Parametern Wechselwirkungen bestehen, verlieren die Ergebnisse der Fenstergröße 9 an Aussagekraft.

Am Beispiel des Motivs Spiel sollen die Experimente 1 und 2 mit der dafür optimierten Fenstergröße 23 durchgeführt werden. Abbildung 28 zeigt, dass alle Werte für das jeweils beste Ergebnis einer Serie sehr nah beieinander liegen. Die besten Resultate ergeben sich in diesem Beispiel bei einer Anhäufung mit Minfilter, SSD und ohne das Verfahren von Birchfeld und Tomasi (interval).

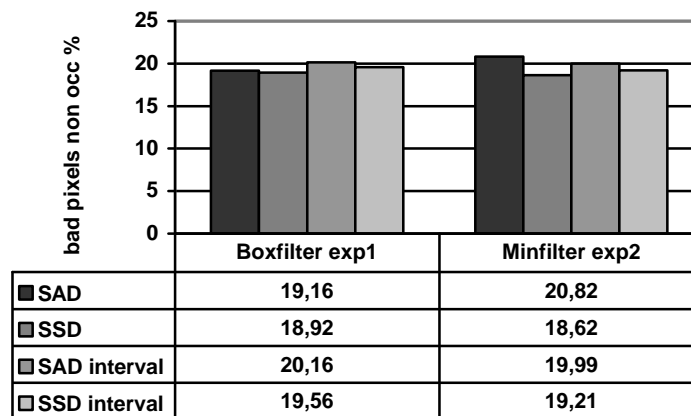


Abb. 28: Ergebnisse der Experimente 1 und 2 für das Motiv Spiel bei optimierter Fenstergröße [A9]

Eine positive Wirkung dieses Verfahrens kann anhand der Experimente 1 und 2 nicht nachgewiesen werden, jedoch kann es motivabhängig der Optimierung dienen. SSD liefert tendenziell bessere Werte als SAD. Minfilter und Boxfilter bringen etwa gleich gute Leistungen. Da die besten Ergebnisse meist mit Truncation = 50 oder ohne Truncation entstehen, wobei in diesem Fall eine programminterne Begrenzung der Matchingkosten auf 1000 durchgeführt wird, wurde mit Zwischenwerten von 100 bis 500 experimentiert. Gegebenenfalls sind so leichte Verbesserungen möglich.

5.4.2. Optimierung von Scanline Optimization

Experiment 5 zeigt, dass entsprechend Tabelle A.5. Bestleistungen für das Motiv Spiel bei einer optischen Glätte von 200 auftreten, und alle weiteren Motive Werte zwischen 200 und 1000 benötigen. In Experiment 3 wird jedoch für SO die optische Glätte 20 vorgegeben, um verschiedene Truncation-Werte und das Verfahren von Birchfeld und Tomasi zu testen. Analog zu den Experimenten 1 und 2 soll nun Scanline Optimization in Experiment 3 mit optimierten Parametern getestet werden, also mit optischer Glätte 200.

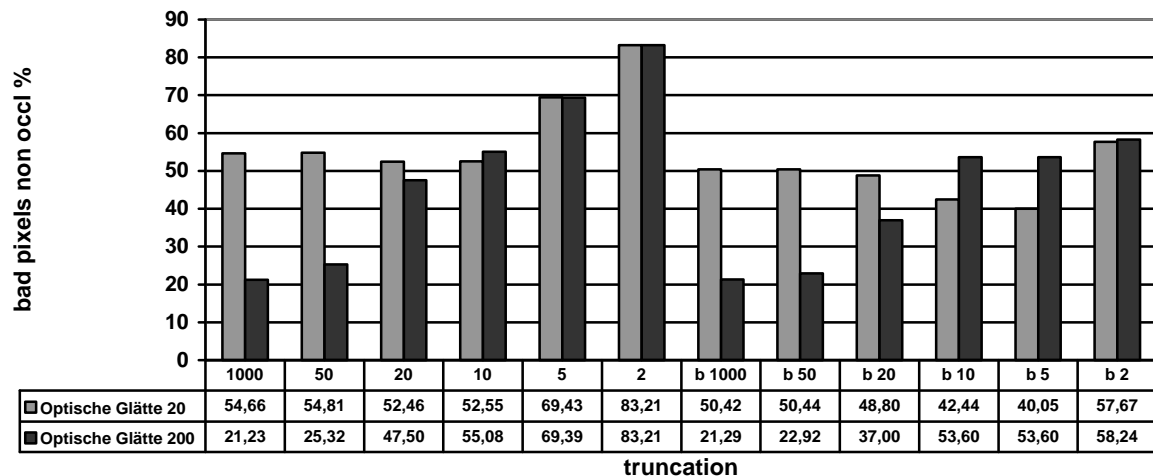


Abb. 29: Scanline Optimization, Motiv Spiel:

Abhängigkeit des Prozentsatzes der nicht verborgenen Pixel von der optischen Glätte und den Truncation-Werten
 b = Benutzung des Birchfeld-Tomasi-Verfahrens [A9]

Wie Abbildung 29 zeigt, ergab sich bei der optischen Glätte von 20 eine Bestleistung von 40,05% der nicht verborgenen Pixel bei Truncation = 5. Ein Erhöhen der optischen Glätte ergibt beste Resultate für unbegrenzte Matchingkosten (ohne Truncation) und ohne Birchfeld-Tomasi-Verfahren. Der Fehlerwert der nicht verborgenen Pixel beträgt 21,23 % und liegt noch unter dem besten Wert aus Experiment 5 mit 21,53%. Anhand dieses Beispiels wird deutlich, wie wichtig eine optimale Abstimmung der Parameter ist.

5.4.3. Optimierung von Simulated Annealing

Das Vergrößern der optischen Glätte führte zu keiner Verbesserung, jedoch wurde die Variable der Temperatur, die den Wert 200 hatte, verändert. Wird der Wert beispielsweise auf 50 gesenkt, steigert sich die Leistung des Algorithmus für das Motiv Spiel deutlich. Zusätzlich wurde die Anzahl der Iterationen, die ursprünglich 500 betrug, auf bis zu 10000 erhöht. Zwar vervielfachen sich die Bearbeitungszeiten entsprechend, die Leistungssteigerung ist jedoch enorm.

Temperatur	200	50	50	50	50
Iterationen	500	500	1000	5000	10000
bad pixels non occ %	43,43	27,93	21,48	14,10	13,09

Ergebnisse für Simulated Annealing bei optischer Glätte 50

Vergleicht man die neue Bestleistung des Verfahrens mit den anderen Ergebnissen des Motivs Spiel, so stellt Simulated Annealing nun das zweitbeste Verfahren nach Graph Cuts dar.

5.4.4. Dynamic Programming und Graph Cuts

Sowohl Dynamic Programming als auch Graph Cuts wurden in den Standardversuchen mit einer Vielzahl von Parametern ausführlich getestet. Eine Veränderung der Werte brachte daher keine weitere Verbesserung.

5.5. Verkleinern des Disparitätsschrittes und Subpixelverfeinerung

Anhand des Motivs Chaos_tief sollen die Wirkungen von kleineren Disparitätsschritten und Subpixelverfeinerung erforscht werden. Das Motiv wurde wegen der Tiefenverläufe der Objekte ausgewählt, die sich für die visuelle Bewertung gut eignen. Alle bisherigen Experimente wurden mit Disparitätsschritt 1 durchgeführt. Multipliziert man den Disparitätsschritt mit dem Wert der Variablen `disp_scale`, der in den selbst erstellten Motiven 12 beträgt, erhält man die Skalierung der Helligkeitswerte. Werden feinere Abstufungen gewünscht, so muss die Variable `disp_step` auf einen Wert $1/N$ gesetzt werden, N muss eine ganze Zahl sein.

Da die selbst erstellten Groundtruth-Karten nur eine Genauigkeit in der Größenordnung von $d = 1$ aufweisen, können Fehlerwerte, die die Groundtruth-Karte als Referenz benutzen, keine aussagefähigen Werte liefern.

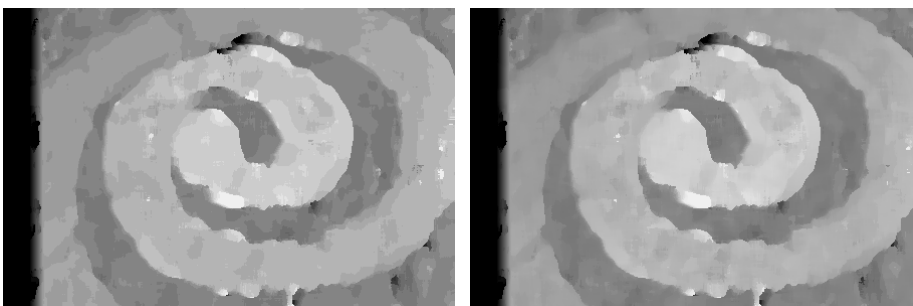


Abb. 30 a, b: Tiefenkarte des Motivs Chaos_tief, Experiment 4, SAD 13 [A9]

links: `disp_step = 1`; ohne Subpixelverfeinerung

rechts: `disp_step = 0,25`; mit Subpixelverfeinerung

Die Qualitätssteigerung ist jedoch visuell erkennbar und kann bei der Rekonstruktion der Szene störende Stufungen an schräg stehenden Objekten vermeiden und fließende Tiefenübergänge erzeugen, sowie eine exaktere Abschätzung der Tiefe ermöglichen.

Falls die Unterschiede in der gedruckten Fassung nicht erkennbar sind, können die Bilder auf einer der beiliegenden CDs betrachtet werden. Details über die Verteilung der Helligkeitswerte liefern Histogramme der oben dargestellten Tiefenkarten:

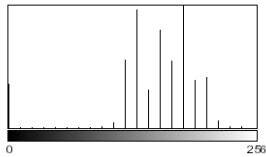


Abb. 31a: disp_step = 1

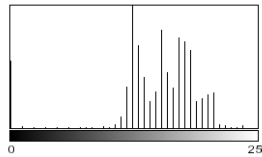


Abb. 31b: disp_step = 0,5

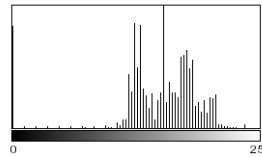


Abb. 31c: disp_step = 0,25

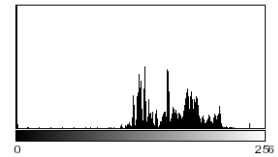
Abb. 31d: disp_step = 0,25
mit Subpixelverfeinerung

Abb. 31 a-d: [A9]

6. Diskussion

6.1. Beurteilung der einzelnen Verfahren

In den folgenden Kapiteln werden die einzelnen Verfahren bezüglich der Fehlerstatistiken, des visuellen Eindrucks der Tiefenkarten und der benutzten Parameter analysiert.

6.1.1. Lokale Methoden

Anhand der Tabelle A.4. ist zu erkennen, dass in Experiment 1 die Anhäufung mit Boxfilter der Anhäufung mit Minfilter leicht überlegen ist. Vergleicht man jedoch die Leistungen aus Experiment 4, so fällt auf, dass in diesem Fall Minfilter die besten Ergebnisse liefert. Dabei ist zu beachten, dass die Fenstergrößen, die zu diesen Werten geführt haben, im Allgemeinen deutlich größer sind, als die festgesetzte Standardfenstergröße 9 aus Experiment 1, meistens 23 bis 29. Erfahrungsgemäß benötigen komplizierte Bilder größere Fenster als einfache. Somit sind die Ergebnisse aus Experiment 1 nicht sehr aussagekräftig, da die Fenstergröße zwar für die relativ einfach gestalteten Bilder der Autoren optimiert ist, aber offensichtlich nicht für die selbst erstellten Stereobilder.

Die Untersuchungen in Kapitel 5.4.1. führen zu der Folgerung, dass bei optimaler Fenstergröße Minfilter und Boxfilter etwa gleiche Ergebnisse bringen und das Birchfeld-Tomasi-Verfahren kaum erkennbare Verbesserungen ergibt. Die optimale Fenstergröße ist meist ein Kompromiss, da zu große Fenster zu Unschärfe und Verschwimmen der Umrisse führen, während zu kleine Fenster texturlose Stellen oder andere schwierige Bildbereiche nicht mit einem Helligkeitswert belegen können.

In den Experimenten 1 und 2 liefern laut Tabelle A.5. Parameterkombinationen ohne Truncation oder mit Truncation zwischen 20 und 50 die besten Ergebnisse. In Experiment 4 zeigt sich eine breite Streuung der Bestleistung über alle Parameter, jedoch mit der Tendenz zur maximalen Fenstergröße beziehungsweise den meisten Iterationen.

Die Bestleistung liegt je nach Motiv zwischen 4% und 60% für die nicht verborgenen Pixel. Die lokalen Methoden können gut durchstrukturierte einfache Objekte gut wiedergeben, scheitern jedoch an schlecht texturierten Objekten, welche oft nur am Rand richtig dargestellt werden. Das Innere besteht zum Teil aus Stellen verschiedenster Helligkeit, so dass das Ablesen eines konkreten Grauwertes und somit eine konkrete

Entfernungsbestimmung für jedes Bildteil nicht möglich sind. Gegenstände werden meist nicht als Ganzes erfasst. Umrisse entsprechen oft nicht genau den Gegebenheiten, was unter Anderem auf den in [1] beschriebenen Effekt des "Foreground Fattening" zurückzuführen ist.

Die Bildgebung in den lokalen Methoden besteht aus viereckigen Flecken entsprechend der Fenstergröße. Dadurch wirkt die Tiefenkarte unruhig, es entstehen keine glatten Flächen. Das Diffusionsverfahren unterstützt zwar eine flächige Gestaltung, jedoch leidet das entstehende Bild unter einem starken Verschwimmen der Konturen. Das Membranverfahren versucht, dieses Verschwimmen zu unterdrücken, jedoch entstehen dabei wieder unruhige Strukturen.

Bei einfachen Motiven können die lokalen Methoden mit der Qualität der globalen Methoden mithalten. Liegt jedoch ein schwieriges Motiv vor, werden die Unterschiede deutlich.

6.1.2. Dynamic Programming

Dynamic Programming (Experimente 3 und 5) liefert relativ gute Bilder. Störend wirkt sich die starke Querstreifenbildung aus, die in Kombination mit bestimmten Parametern teilweise zu vollkommen unbrauchbaren Tiefenkarten führt. Schwach strukturierte Stellen können trotzdem einheitlich dargestellt werden. Die Umrisse der Gegenstände sind gut zu erkennen, oft ist jedoch kein eindeutiger Helligkeitswert ablesbar, da die Objekte meist aus Streifen verschiedener Helligkeit zusammengesetzt werden.

Das Verfahren von Birchfeld und Tomasi verbessert die Ergebnisse. Die optimalen Werte der Truncation liegen in Experiment 3 zwischen 10 und 50. Kleinere Werte führen zu glatteren Bildern. In Experiment 5 entstehen die besten Tiefenkarten bei Glattheitswerten zwischen 50 und 500, dabei sind die Bestleistungen auf alle Werte der Occlusion Cost, 20, 50 und 80, gleichmäßig verteilt. Die besten Prozentwerte der nicht verborgenen Pixel schwanken zwischen 5% und 56% in Experiment 3 und zwischen 2% und 44% in Experiment 5.

6.1.3. Scanline Optimization

Scanline Optimization (Experimente 3 und 5) liefert nicht ganz so gute Ergebnisse wie Dynamic Programming, einige Bilder sind ebenfalls wegen der starken Streifenbildung

unbrauchbar. Die Ganzheit der Objekte wird nicht immer überzeugend wiedergegeben. Kanten werden gut erkannt, aber schwach strukturierte Bereiche werden oft in Strichen und Punkten aller Helligkeiten wiedergegeben. Bei idealer optischer Glätte (Kapitel 5.4.2.) werden Truncation-Werte von etwa 50 benötigt. Zur Optimierung ist das Birchfeld-Tomasi-Verfahren hilfreich. Experiment 5 zeigt, dass Glattheitswerte zwischen 200 und 1000 beste Resultate ergeben.

Experiment 3 liefert prozentuale Fehlerwerte von 7% bis 57%, in Experiment 5 entstehen Werte zwischen 2% und 44% für die besten Bilder einer Serie.

Dynamic Programming und Scanline Optimization sind ausgesprochen interessante Verfahren, da sie die Schnelligkeit der lokalen Methoden mit der Qualität der globalen Methoden kombinieren.

6.1.4. Simulated Annealing

Simulated Annealing liefert im Vergleich mit anderen globalen Verfahren eher dürftige Ergebnisse. Charakteristisch für die Bildgebung in den Tiefenkarten sind mosaikähnliche Flecken verschiedener Größe. Dadurch können die Bildelemente bis zur Unkenntlichkeit zerstückelt werden. In Experiment 5 ergab sich die beste Qualität bei einer optischen Glätte von etwa 500.

Durch Variation der Parameter, wie in Abschnitt 5.4.3. beschrieben, konnte die Leistung deutlich verbessert werden. Anscheinend benötigen schwierige Bilder mehr Iterationen. Nach Szeliski [1] löst Graph Cuts das gleiche Optimierungsproblem effizienter, da nicht wie bei SA in jedem Schritt der Wert jedes Pixels geändert werden muss. Der Fehler der nicht verborgenen Pixel liegt motivabhängig zwischen 2% und 53%.

6.1.5. Graph Cuts

Graph Cuts liefert im Allgemeinen die besten Bilder. Objekte werden im Ganzen erfasst, nicht nur an den Kanten. Die Darstellung ist flächig, trifft aber leider manchmal nicht die wirklichen Konturen. Das Ablesen von Helligkeitswerten aus der Tiefenkarte im Hinblick auf eine Rekonstruktion der dargestellten Bildelemente ist wegen der wenigen großen Flächen besonders einfach und effektiv. Im Gegensatz zu allen anderen Methoden entstehen bei hohen Kontrasten innerhalb eines Objektes fast keine das Ergebnis verfälschenden Helligkeitsabweichungen.

Graph Cuts zeigt Bestleistung für eine Truncation von 10 bis 20 (Experiment 3), Glattheitswerte von etwa 500 (Experiment 5), und Penalty-Werten von 4 (Experiment 6). Die prozentualen Fehler für optimierte Tiefenkarten liegen in manchen Experimenten unter 1% und erreichen maximal 42% (Experiment 3), 32% (Experiment 5) bzw. 33% (Experiment 6).

Das Verfahren überzeugt nicht nur bei der Bewertung mit prozentualen Fehlern, sondern auch bei visueller Beurteilung. Die besonders flächige Darstellung trennt deutlich einzelne Bildteile voneinander ab. Da es nur wenige Grenzverläufe der Flächen gibt, wirkt es manchmal störend, wenn eine solche Grenze mitten durch ein Objekt verläuft. Die Abstimmung der Truncation wie auch der optischen Glätte sind Kompromisse zwischen dem Detailreichtum des Bildes, der jedoch zur Zerstückelung der Grauwertflächen führt, und einer flächigen klaren Darstellung, die den Verlust der Details zur Folge hat.



Abb. 32 a, b, c, d: Tiefenkarten für das Motiv Stern, Graph-Cut-Verfahren mit steigender optischer Glätte [A9]

6.2. Vergleich aller Verfahren

Die Leistung aller Verfahren ist sehr stark von der Art des zu matchenden Bildes abhängig. Ein Bild, das in allen Bereichen gut, aber nicht zu kontrastreich durchstrukturiert ist und wenige gestaffelte Bildebenen senkrecht zur optischen Achse aufweist, liefert optimale Voraussetzungen. Einige Motive enthalten jedoch schwach strukturierte Stellen. Manche Methoden sind nicht in der Lage, solche Stellen in einem relativ einheitlichen Grauton wiederzugeben. Nur Dynamic Programming und Graph Cuts erlauben eine ganzheitliche Darstellung.

Weitere gestalterische Schwachstellen vieler Methoden zeigen sich an Stellen mit starken Helligkeitsunterschieden, die oft als Tiefensprung interpretiert werden. Ein Objekt, dem eigentlich ein einheitlicher Grauwert zugeordnet werden sollte, wird zerstückelt und ist nicht mehr als Ganzes zu erkennen. Die starken Kontraste werden offensichtlich als Kanten interpretiert und entsprechend dargestellt, so dass durch die verschiedenen Grauwerte in einem Objekt keine exakte Tiefenabschätzung mehr möglich ist. Eine ausgeprägte Strukturierung in einem Motiv findet sich auch in der Tiefenkarte wieder. Bei wirklich schwierigen Motiven liefern daher alle Methoden unbefriedigende Ergebnisse.

Eine starke Beeinträchtigung der Leistung ist somit sowohl bei strukturschwachen als auch bei sehr stark texturierten Motiven zu finden.

Besonders schräge Flächen, die sich in der Tiefe über mehrere Disparitätseinheiten erstrecken, bereiten allen Methoden erhebliche Probleme. Der Untergrund der Motive Spiel oder Würfel oder die Bildelemente im Motiv Pfeil demonstrieren die Tatsache, dass teilweise nur Rauschen dargestellt wird oder der Tiefenverlauf wellenförmig gestaltet wird. Das schwierigste Stereobildpaar in diesen Tests war das Motiv Glühbirnen, da es aus unstrukturierten Flächen, hohen Kontrasten, Rundungen, Tiefenverläufen und den Reflexe verursachenden Materialien Glas und Metall besteht. Da jedoch ein Stereoalgorithmus, der beispielsweise Bilder aus einem Roboter oder Fahrzeug auswerten soll, *jedes* Hindernis erkennen muss, egal welcher Art es ist, ist es durchaus sinnvoll, ein solch kompliziertes Motiv in die Testserie mit aufzunehmen.

Vergleicht man die Tiefenkarten aller Verfahren bezüglich der Helligkeitsabstufungen für bestimmte Objekte, so fällt auf, dass jedes der Verfahren in der Lage ist, exakt den richtigen Wert zu treffen. Das Finden der richtigen Disparität für den jeweiligen Bildteil ist für keines der Verfahren ein Problem, in jeder Tiefenkarte lässt sich mit "gutem Willen" der korrekte Helligkeitswert finden. Lediglich die Art der Darstellung, die teilweise an moderne Kunst erinnert, ist oft sehr eigenwillig und fehleranfällig, wie in den folgenden Beispielen dargestellt.

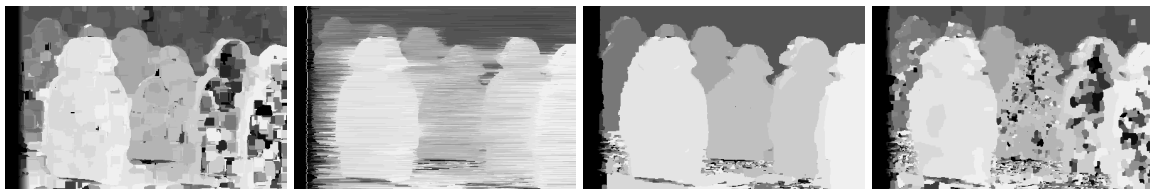


Abb. 33 a, b, c, d: Tiefenkarten für das Motiv Spiel. Typische Ergebnisse verschiedener Algorithmen. Von links nach rechts:

WTA mit Minfilter, Dynamic Programming, Graph Cuts, Simulated Annealing [A9]

Mit einer gewissen menschlichen Gutmütigkeit kann man aus solchen Tiefenkarten aussagefähige Folgerungen über die Tiefensituation der dargestellten Szene ziehen. Unklar bleibt jedoch, ob auch ein Computerprogramm in der Lage wäre, diesen Karten sinnvolle Informationen zu entnehmen.

Wirklich für Tiefenschätzungen zu gebrauchen ist anscheinend nur das Graph Cut Verfahren, da alle anderen Verfahren durch Flecken, Streifen und andere Artefakte klare Tiefenzuordnungen erschweren.

Vergleicht man die Verfahren im Hinblick auf die Zeiten, die zur Erstellung einer Tiefenkarte und der dazu gehörenden Textdatei mit Fehlerwerten benötigt werden, fällt auf, dass lokale Verfahren für Bildgrößen von $450 * 300$ Pixel auf einem Pentium III

Rechner mit 1 GHz Taktfrequenz etwa 1 bis 2 Sekunden benötigen, bei iterativen Verfahren bis 30 Sekunden. Dynamic Programming und Scanline Optimization erstellen ebenfalls Tiefenkarten innerhalb weniger Sekunden, da die Optimierung nur entlang einer Linie stattfindet. Graph Cuts und Simulated Annealing optimieren alle Bildpunkte gleichzeitig und benötigen dafür Zeiten zwischen 10 und 20 Minuten. Alle Zeiten finden sich in Tabelle A.6. Findet eine Verkleinerung des Disparitätsschrittes oder Subpixelverfeinerung statt, verlängern sich die Zeiten entsprechend.

Die lokalen Methoden sowie Dynamic Programming und Scanline Optimization könnten mit ihren schnellen Bearbeitungszeiten für echtzeitfähige Experimente taugen, wie z.B. das Erkennen von Hindernissen mit den "Augen" eines sich bewegenden Roboters.

6.3. Bewertung des Prediction Error

In Kapitel 5.2. wurde die Leistung des Prediction Error diskutiert. Abschließend stellt sich die Frage, wie dieses so andersartige Fehlermaß beurteilt werden kann. Mit dem Prediction Error wird die Fähigkeit einer Tiefenkarte bewertet, ein korrespondierendes Stereobild vorherzusagen, jedoch wird nicht die Tiefenkarte selbst bewertet. Da für die Rekonstruktion einer dreidimensionalen Situation die Tiefenkarte Informationen über Entfernungen liefern soll, muss sie selbst zur Bewertung herangezogen werden. Statt des Fehlers zwischen errechnetem und echtem Stereobild sollte daher der Fehler zwischen Groundtruth-Karte und Tiefenkarte ermittelt werden.

Wenn die Erstellung von Tiefenkarten der Rekonstruktion dienen soll, ist der Prediction Error als Fehlermaß ungeeignet. In anderen Bereichen, wie der Erschaffung neuer Ansichten, mag er seine Berechtigung haben. In dieser Arbeit sollen jedoch zur quantitativen Beurteilung der Stereoverfahren andere Fehlermaße herangezogen werden.

6.4. Bewertung der von den Autoren Scharstein und Szeliski veröffentlichten Bilder und Ergebnisse

Durch die Verwendung der ausgesprochen einfach gestalteten Bilder Sawtooth, Venus und Map entstehen verfälschte Erkenntnisse über die Wirkung der Parameter. Da die Fehlermaße ausgesprochen gering sind, ergeben sich z.B. Trugschlüsse über die Konvergenz von Iterationen oder die optimale Fenstergröße.

Es ist nicht nachvollziehbar, warum die Autoren nur besonders einfache Motive zum Testen der Verfahren heranziehen. Schwierige Motive sind wesentlich besser in der Lage, Vor- und Nachteile des jeweiligen Verfahrens aufzuzeigen und Verbesserungen durch optimierte Parameter werden anhand der Fehlermasse deutlich sichtbar. Realitätsnahe Motive erfordern oft andere Abstimmungen der Parameter, z.B. größere Fenster in den lokalen Methoden.

Daher ist es absolut unverständlich, dass die Programme von den Autoren nicht in schwierigen Situationen, d.h. realitätsnahen Bildern getestet wurden, bzw. keine solchen Testergebnisse veröffentlicht wurden. Jede Erfindung, jedes Verfahren oder jedes Produkt durchläuft normalerweise Härtetests, um Schwachstellen und Grenzen aufzuzeigen.

6.5. Beurteilung der Implementierung

Anhand der in das Programm integrierten Fehlermessung ist es möglich, verschiedene Verfahren quantitativ zu bewerten. RMS-Fehler und prozentualer Fehler liefern dabei Ergebnisse, die ziemlich gut mit einer qualitativen visuellen Bewertung der Tiefenkarten korrespondieren.

Eine große Stärke des Programms ist die Gestaltung des "Baukasten-Systems", die es erlaubt, einzelne Parameter zu variieren, während andere konstant gehalten werden. So können detaillierte Erkenntnisse über die Wirkung eines jeden Parameters gewonnen werden. Jedoch beinhaltet genau diese Systematik auch eine Schwierigkeit: Die konstant gehaltenen Parameter sind nicht immer für das jeweilige Motiv optimiert, und die Veränderung dieses Parameters hat unter Umständen Auswirkungen auf die Leistung des eigentlich zu untersuchenden Parameters (Siehe auch Kapitel 5.4.).

In dem Beispiel des Simulated Annealing aus Abschnitt 5.4.3. ergab sich bei einer vorgegebenen Temperatur von 200 die beste Leistung des zu verändernden Parameters optische Glätte bei einem Wert von 200. Senkt man die Temperatur auf 50, verschiebt sich die Bestleistung auf die optische Glätte 50.

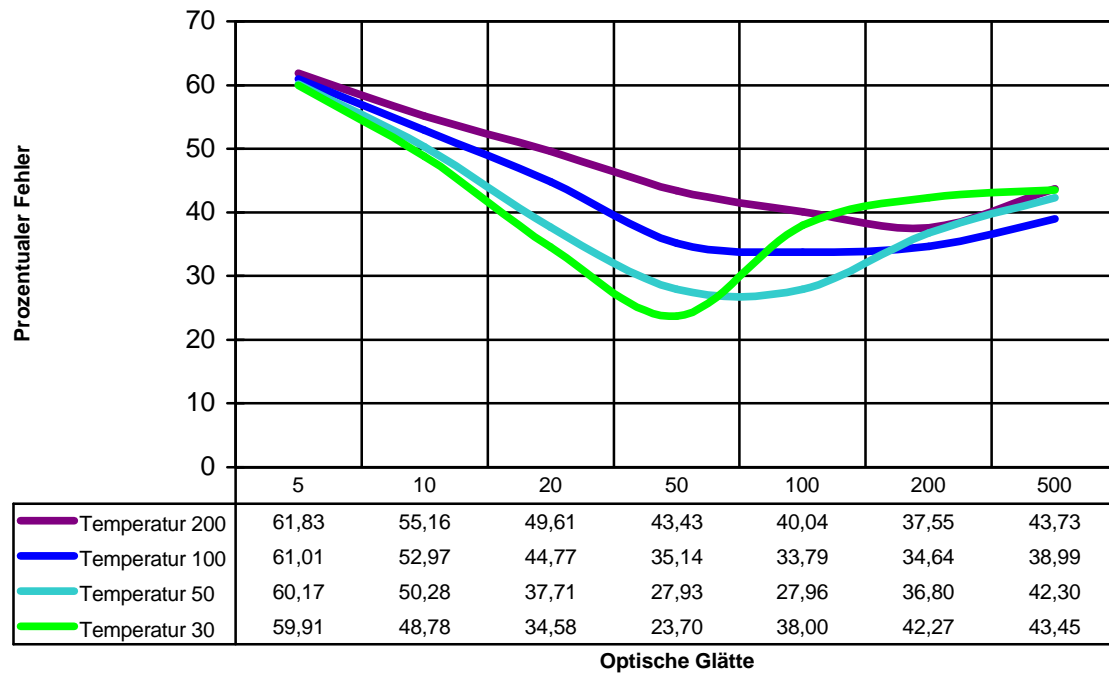


Abb. 34: Verschiebung des Funktionsminimums des prozentualen Fehlers von Simulated Annealing in Abhängigkeit von der optischen Glätte bei Variation der Temperatur [A9]

Da sich die Parameter gegenseitig beeinflussen, genügt es nicht, nur jeweils einen Parameter zu verändern, um Erkenntnisse über seine Wirkung zu erhalten. Um wirklich brillante Ergebnisse zu erzielen, müssen alle Variationen von Parameterkombinationen durchgetestet werden. Durch die vielen Einfluss nehmenden Werte würde sich damit der experimentelle Aufwand vervielfachen. Jedoch hat der Benutzer die vielfältigsten Möglichkeiten, vorgegebene Experimente abzuändern und zu variieren.

7. Zusammenfassung

In dieser Arbeit wurde gezeigt, dass es möglich ist, anhand zweier Stereobilder Informationen über die räumliche Entfernung der Objekte zu erhalten. Die Ungenauigkeiten der hier veröffentlichten Werte sind zu einem großen Teil auf die Verschiebung der Kamera, auf Unsicherheiten bezüglich der Eigenschaften des optischen Systems und Fehlerfortpflanzung bei der Berechnung zurückzuführen.

Die Implementierung von Scharstein und Szeliski wird beispielsweise in [24] genutzt, um Bilddaten aus Aufnahmen im Straßenverkehr zu beurteilen. Das Interesse der Forschung besteht in der Entwicklung neuer auf Stereovision basierender Techniken für Anwendungen im Bereich der intelligenten Fahrzeuge.

Um Sicherheit zu garantieren, muss gewährleistet werden, dass sich auf Gebieten mit solchen Fahrzeugen keine weiteren Verkehrsteilnehmer befinden. Hindernisse oder Personen können anhand der Tiefenkarten detektiert werden. Dabei werden die von Scharstein und Szeliski implementierten Verfahren mit weiteren Verfahren verglichen, die besonders im Hinblick auf Echtzeitfähigkeit herausragende Resultate liefern. Mit Hilfe der Stereoverfahren können somit praktische Lösungen für vielfältige Aufgabenstellungen gefunden werden. Eine Auswahl von Veröffentlichungen von Szeliski zu verschiedenen Themen der Computer Vision findet sich unter [25].

In der vorliegenden Arbeit wurden verschiedene Stereoverfahren getestet und eine Reihenfolge ihrer Leistungen aufgestellt, die der der Autoren ähnelt. Die relativen Leistungen der Verfahren sind jedoch sehr von der Art des Motivs abhängig.

Als mit Abstand bestes Verfahren hat sich das Graph Cut Verfahren herausgestellt, dessen Laufzeiten jedoch keine echtzeitfähigen Anwendungen zulassen. Von den schnellen Verfahren überzeugt Dynamic Programming am besten. Wie auch in [1] diskutiert, ist die Wirkung der Parameter je nach Kombination und Motiv sehr verschieden. Daher muss für gute Ergebnisse viel Aufmerksamkeit in das Testen der Parameter gelegt werden.

Nur einige wenige der aktuellen Stereoverfahren konnten hier beschrieben und bewertet werden. Es existiert jedoch eine Vielzahl von weiteren Verfahren, die ähnlich interessante Resultate liefern.

Leider konnte in dieser Arbeit der nächste Schritt, die Rekonstruktion der in den Stereobildern dargestellten Objekte, nicht mehr in die Aufgabenstellung integriert werden. Da in den Tiefenkarten für jeden Bildpunkt eine Helligkeit vorhanden ist, aus der die

Entfernung ermittelt werden kann, kann die ursprüngliche Aufnahmesituation nachgebildet werden.

Eine Art pseudodreidimensionale Darstellung ergibt sich beispielsweise, wenn eine Tiefenkarte proportional zu ihren Helligkeiten in die entsprechende Höhe der Z-Achse abgebildet wird. Die Proportionen bezüglich der räumlichen Tiefe stimmen daher nicht.



Abb. 35a: Tiefenkarte des Motivs Chaos_tief [A9]

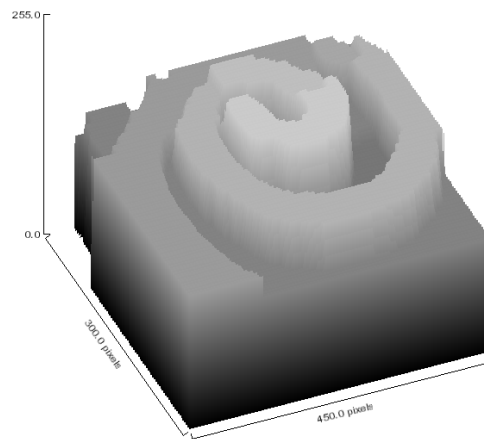


Abb. 35b: Surface Plot der Tiefenkarte [A9]

Jedoch soll dieses Beispiel, das als "Surface Plot" mit dem Programm ImageJ entstanden ist, die visuellen Möglichkeiten der Rekonstruktion andeuten.

Anhang

A.1.a. Entstehung des Plugins entzerrung_.java

Aus den Ist- und Soll-Werten der Referenzpunkte im Bild wurde eine dreidimensionale Look Up Table erstellt. Je vier Werte gehören zusammen: x_{soll} , y_{soll} , x_{ist} , y_{ist} . Diese Werte werden in x-Richtung und in y-Richtung je neun Mal benötigt. Der Reihe nach wird jeder Punkt des Ausgangsbildes, also des neu zu erschaffenden entzerrten Bildes, bearbeitet. In [23] S. 279, werden die Eigenschaften von Vorwärts- und Rückwärtsabbildung beschrieben. Die hier benutzte inverse Abbildung bildet die Koordinaten eines Punktes des Ausgangsbildes auf das zu entzerrnde Eingangsbild ab. Da der Punkt dort nicht exakt auf ein Pixel abgebildet wird, muss aus den umliegenden Werten interpoliert werden.

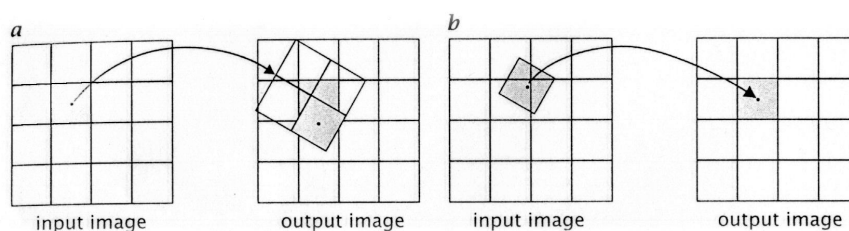


Abb. 36a: Vorwärtsinterpolation, 36b: Rückwärtsinterpolation [A8]

Die Vorwärtsinterpolation, bildet einen Punkt des Eingangsbildes auf das Ausgangsbild ab. Der Wert des Pixels im Eingangsbild muss auf mehrere Pixel des Ausgangsbildes verteilt werden. Damit keine Löcher gerissen werden, kann der Anteil der Fläche des Eingangspixels, welcher auf das Ausgangspixel fällt, als Wichtungsfaktor genommen werden. Jedes Pixel des Ausgangsbildes erhält die gewichteten Farbwerte verschiedener Eingangspixel. Die inverse Abbildung hat den Vorteil der einfacheren Vorgehensweise, wie z.B. bilineare Interpolation.

Die Pixel des Eingangsbildes werden in ein eindimensionales Array eingelesen (Zeile 31 des nachfolgend abgedruckten Quelltextes). Die jeweiligen x- und y- Werte können mit Hilfe des Restoperators bzw. durch Teilen durch die Bildweite erhalten werden (Z. 138-139).

Teilen der Koordinaten durch die entsprechenden Seitenlängen eines Soll-Kästchens ergibt die Indices m und n der Look Up Table für die Bestimmung des Referenzvierecks, in dem das Pixel liegt (Z. 143-144).

Die Skalierungsfaktoren `scale_x` und `scale_y` bestimmen die relative Position des Pixels innerhalb des Referenzvierecks (Z.150-151).

Die Hilfspunkte H1 und H2 skalieren die Strecken des verzerrten Referenzkästchens entsprechend den Proportionen aus dem entzerrten Bild (Z. 153-156).

Die Berechnung von `x_ist` ergibt sich aus der Mittelung der x-Werte beider Punkte, indem der x-Wert jedes Punktes mit der Skalierung des Abstandes des anderen Punktes multipliziert wird und beide Werte addiert werden (Z. 158).

Analog wird mit Hilfe von H3 und H4 `y_ist` berechnet (Z. 160-165). Somit sind `x_ist` und `y_ist` des zu bearbeitenden Punktes bekannt. Für diesen Punkt muss nun ein Farbwert definiert werden, der in das entzerrte Bild geschrieben wird.

Da `x_ist` und `y_ist` zum großen Teil zwischen mehreren Pixeln liegen, müssen diese Pixel bestimmt und ihre Farbwerte entnommen werden. Dazu werden aus dem Array der verzerrten Bildpunkte mit den Funktionen `Math.floor ()` und `Math.ceil ()` die Farbinformationen für die umliegenden Pixel bestimmt (Z. 170-173). Alle drei Grundfarben R, G und B sind in einer Variablen vereinigt und müssen vor der Interpolation getrennt werden (Z. 177-191).

Für eine bilineare Interpolation, die die Farben entsprechend ihres Abstandes gewichtet, wird erst jeder Farbwert mit dem seines in x-Richtung liegenden Nachbarn interpoliert, indem er mit dem Abstand des Nachbarn (Berechnung des Abstandes: Z. 195-198) gewichtet und addiert wird (Z. 202-208). Die so entstandenen zwei Hilfsfarbwerte werden auf die gleiche Weise in y-Richtung interpoliert (Z. 210-212). Es entstehen die Farbwerte r, g, b, welche in der Variablen W vereinigt werden (Z. 214). Dieser Wert kann nun an die Position `x_soll, y_soll` eingetragen werden (Z. 218).

A.1.b. Programmcode

```

1 // Programm entzerrt kissenförmige Verzeichnung eines Bildes
2 // Plugin für ImageJ
3 // Sylvia Lange 12.2.2005
4
5 import ij.*;
6 import ij.gui.*;
7 import ij.plugin.filter.PlugInFilter;
8 import ij.process.*;
9 import java.awt.*;
10 import java.lang.Math.*;
11
12 public class entzerrung_ implements PlugInFilter {
13
14
15     public int setup ( String arg, ImagePlus imp ) {
16         if ( arg.equals ("about") ) {
17             showAbout ();
18             return DONE; }
19         return DOES_RGB;

```

A.2. Vergleich der aus den Tiefenkarten berechneten Entfernungen A mit den tatsächlichen Entfernungen B

Motiv Spiel	Helligkeitswert	Disparität	Entfernung A	Entfernung B
Figur lila	240	40	69,83	69
Figur blau	228	39	71,43	70
Figur rot	204	37	74,91	71
Figur orange	180	35	78,78	75
Figur gelb	168	34	80,89	77
Figur grau	132	31	88,03	86
Hintergrund	96	28	96,69	95
	84	27	100,01	
	72	26	103,58	

Motiv Karten	Helligkeitswert	Disparität	Entfernung A	Entfernung B
Herz König	240	50	57,29	69
Karo Bube	186	45,5	62,25	77
Pik Dame	108	39	71,43	86
Hintergrund	96	38	73,13	92

Motiv Kiwi	Helligkeitswert	Disparität	Entfernung A	Entfernung B
halbe Kiwi vorne	216	38	73,13	74
	204	37	74,91	
halbe Kiwi links	180	35	78,78	77
	168	34	80,89	
Kiwi rechts	156	33	83,12	81
Kiwi hinten	132	31	88,03	87
Hintergrund	96	28	96,69	95
	84	27	100,01	

Motiv Laub	Helligkeitswert	Disparität	Entfernung A	Entfernung B
Blatt vorne	228	39	71,43	73
	216	38	73,13	
	204	37	74,91	
Blatt links	168	34	80,89	82
	156	33	83,12	
Blatt rechts	96	28	96,69	97
Hintergrund	60	25	107,44	106

Motiv Zickzack	Helligkeitswert	Disparität	Entfernung A	Entfernung B
vorne	192	36	76,79	74
mitte	144	32	85,50	83
hinten	84	27	100,01	97

Motiv Glühbirnen	Helligkeitswert	Disparität	Entfernung A	Entfernung B
Glühbirne vorne	216	38	73,13	75
	204	37	74,91	
	192	36	76,79	
	180	35	78,78	
Glühbirne links	132	31	88,03	84
	120	30	90,72	
Hintergrund	60	25	107,44	99
	48	24	111,62	

Motiv Würfel	Helligkeitswert	Disparität	Entfernung A	Entfernung B
6	180	45	62,86	67
3	156	43	65,45	71
4	132	41	68,30	74
5	108	39	71,43	78
2	96	38	73,13	82
Hintergrund	36	33	83,12	95
	24	32	85,50	

Motiv Jeans	Helligkeitswert	Disparität	Entfernung A	Entfernung B
rechts vorne	228	39	71,43	71
	216	38	73,13	
	204	37	74,91	
links vorne	168	34	80,89	78
	156	33	83,12	
links mitte	144	32	85,50	83
	132	31	88,03	
Hintergrund	120	30	90,72	88
	108	29	93,60	

Motiv Pfeil	Helligkeitswert	Disparität	Entfernung A	Entfernung B
Pfeil vorne	240	45	62,86	69
Pfeil mitte	192	41	68,29	75
Pfeil hinten	156	38	73,13	83
Hintergrund	132	36	76,79	86
	120	35	78,78	

Motiv Stern	Helligkeitswert	Disparität	Entfernung A	Entfernung B
Stern	204	42	66,84	69
Hintergrund	108	34	80,89	86
	96	33	83,12	

Motiv TigerTest	Helligkeitswert	Disparität	Entfernung A	Entfernung B
Frosch	168	34	80,89	71
Bär	144	32	85,50	75
Tiger	132	31	88,02	80
Hintergrund	96	28	96,69	88

Motiv Papier	Helligkeitswert	Disparität	Entfernung A	Entfernung B
Vordergrund	222	38,5	72,27	73
	216	38	73,13	
Hintergrund	114	29,5	92,14	96
	108	29	93,60	

Motiv Stifte	Helligkeitswert	Disparität	Entfernung A	Entfernung B
Vordergrund	198	36,5	75,84	74
	192	36	76,79	
Hintergrund	102	28,5	95,12	96
	96	28	96,69	
	90	27,5	98,32	

Motiv Teddy	Helligkeitswert	Disparität	Entfernung A	Entfernung B
Nase	192	36	76,79	74
	180	35	78,78	
Ohr	150	32,5	84,29	79
	144	32	85,50	
Hintergrund	78	26,5	101,76	96
	66	25,5	105,47	

Motiv Chaos_flach	Helligkeitswert	Disparität	Entfernung A	Entfernung B
Vordergrund	168	34	80,89	74
Hintergrund	114	29,5	92,14	86

Motiv Chaos_tief	Helligkeitswert	Disparität	Entfernung A	Entfernung B
Vordergrund	228	39	71,43	72
	210	37,5	74,00	
	204	37	74,91	
Verlauf	186	35,5	77,77	77
	174	34,5	79,82	
Hintergrund	132	31	88,03	88
	126	30,5	89,35	

A.3. Fehlerauswertung der einzelnen Motive

Fehlerauswertung für Motiv Spiel

exp1	bad pixels all	non occl	RMS all	RMS non occl	Prediction Err
SAD	38,47	32,87	5,27	4,65	11,24
SSD	38,12	32,40	5,24	4,60	11,02
SAD interval	37,08	31,32	5,29	4,67	11,15
SSD interval	36,59	30,83	5,24	4,60	11,01

exp2	bad pixels all	non occl	RMS all	RMS non occl	Prediction Err
SAD	36,37	31,08	5,29	4,78	10,65
SSD	35,60	30,21	5,29	4,76	10,08
SAD interval	34,03	28,43	5,36	4,83	10,45
SSD interval	33,73	28,07	5,20	4,65	10,09

exp3	bad pixels all	non occl	RMS all	RMS non occl	Prediction Err
DP	24,11	19,91	2,75	2,21	8,46
SO	42,87	40,05	4,40	4,24	7,30
GC	15,53	9,40	3,27	1,72	8,04

exp4	bad pixels all	non occl	RMS all	RMS non occl	Prediction Err
Boxfilter	29,73	23,18	4,46	3,31	9,20
Minfilter	24,20	18,09	4,19	3,49	8,57
Binomialfilter	29,46	22,88	4,64	3,58	10,18
Diffusion	28,78	22,15	4,51	3,38	10,72
Membran	28,78	22,15	4,51	3,38	9,46

exp5	bad pixels all	non occl	RMS all	RMS non occl	Prediction Err
DP	24,82	18,74	3,40	2,13	8,17
SO	28,07	21,53	4,13	2,96	6,50
GC	15,98	8,25	3,33	1,76	6,63
SA	41,46	36,18	5,91	5,40	6,67

exp6	bad pixels all	non occl	RMS all	RMS non occl	Prediction Err
GC	14,64	6,84	3,36	1,66	8,47

Fehlerauswertung für Motiv Karten

exp1	bad pixels all	non occl	RMS all	RMS non occl	Prediction Err
SAD	31,34	23,72	4,66	2,95	20,45
SSD	31,08	23,44	4,67	2,92	20,29
SAD interval	30,42	22,88	4,53	2,83	20,21
SSD interval	29,93	22,29	4,51	2,82	20,23

exp2	bad pixels all	non occl	RMS all	RMS non occl	Prediction Err
SAD	35,57	28,70	4,81	3,28	20,31
SSD	35,32	28,35	4,76	3,21	19,84
SAD interval	31,36	24,18	4,63	3,01	20,05
SSD interval	32,30	25,19	4,66	3,05	19,73

exp3	bad pixels all	non occl	RMS all	RMS non occl	Prediction Err
DP	17,36	13,63	2,21	1,60	15,84
SO	20,07	17,71	2,85	2,51	11,67
GC	5,68	0,90	1,84	0,82	12,01

exp4	bad pixels all	non occl	RMS all	RMS non occl	Prediction Err
Boxfilter	20,22	11,53	4,47	2,20	17,33
Minfilter	22,50	15,21	4,02	2,14	16,97
Binomialfilter	20,91	12,19	4,46	2,14	18,40
Diffusion	19,20	10,35	4,46	2,14	18,97
Membran	19,20	10,35	4,42	2,08	16,61

exp5	bad pixels all	non occl	RMS all	RMS non occl	Prediction Err
DP	17,62	10,00	3,89	1,75	14,32
SO	18,48	9,49	4,28	1,86	11,62
GC	11,71	1,93	3,87	0,86	11,42
SA	21,90	13,25	4,06	1,82	11,79

exp6	bad pixels all	non occl	RMS all	RMS non occl	Prediction Err
GC	15,06	5,77	3,92	1,36	13,91

Fehlerauswertung für Motiv Kiwi

exp1	bad pixels all	non occl	RMS all	RMS non occl	Prediction Err
SAD	40,30	36,64	4,42	3,80	12,90
SSD	39,75	36,05	4,36	3,74	12,78
SAD interval	39,56	35,90	4,38	3,74	12,93
SSD interval	39,13	35,46	4,32	3,68	12,96

exp2	bad pixels all	non occl	RMS all	RMS non occl	Prediction Err
SAD	40,13	36,83	4,37	3,81	12,77
SSD	39,52	36,22	4,31	3,76	12,08
SAD interval	39,10	35,74	4,33	3,77	12,78
SSD interval	38,35	34,98	4,26	3,70	12,20

exp3	bad pixels all	non occl	RMS all	RMS non occl	Prediction Err
DP	33,20	31,74	2,23	2,15	9,32
SO	37,66	35,93	2,65	2,56	6,88
GC	20,28	18,33	1,71	1,61	8,31

exp4	bad pixels all	non occl	RMS all	RMS non occl	Prediction Err
Boxfilter	28,86	24,62	3,66	2,56	9,55
Minfilter	29,78	26,16	3,42	2,71	9,34
Binomialfilter	31,61	27,41	4,06	3,12	10,36
Diffusion	29,74	25,46	3,93	2,94	11,14
Membran	29,74	25,46	3,93	2,94	8,07

exp5	bad pixels all	non occl	RMS all	RMS non occl	Prediction Err
DP	31,81	27,88	3,52	2,58	8,90
SO	34,54	30,55	3,97	3,03	5,99
GC	18,95	13,92	3,08	1,65	6,18
SA	41,28	37,76	4,69	4,01	6,22

exp6	bad pixels all	non occl	RMS all	RMS non occl	Prediction Err
GC	21,62	17,03	3,61	2,59	9,13

Fehlerauswertung für Motiv Laub

exp1	bad pixels all	non occl	RMS all	RMS non occl	Prediction Err
SAD	27,48	22,90	3,13	2,19	10,32
SSD	26,54	21,78	3,05	2,13	10,33
SAD interval	26,99	22,47	2,99	2,05	10,30
SSD interval	26,56	21,91	2,98	2,04	10,33

exp2	bad pixels all	non occl	RMS all	RMS non occl	Prediction Err
SAD	27,12	23,09	3,00	2,18	10,68
SSD	26,31	22,30	2,95	2,10	9,99
SAD interval	24,45	20,14	2,91	1,97	10,52
SSD interval	24,70	20,45	2,87	1,95	10,04

exp3	bad pixels all	non occl	RMS all	RMS non occl	Prediction Err
DP	14,19	12,88	2,03	1,77	7,93
SO	15,75	15,01	2,30	2,11	6,13
GC	5,83	4,76	1,78	1,22	7,28

exp4	bad pixels all	non occl	RMS all	RMS non occl	Prediction Err
Boxfilter	20,11	14,82	3,24	2,15	8,15
Minfilter	14,71	10,32	2,74	1,86	7,88
Binomialfilter	21,15	15,86	3,31	2,17	8,60
Diffusion	20,36	15,05	3,31	2,17	9,11
Membran	20,36	15,05	3,34	2,22	7,09

exp5	bad pixels all	non occl	RMS all	RMS non occl	Prediction Err
DP	15,40	10,35	2,83	1,61	7,43
SO	14,26	8,72	3,11	1,79	5,32
GC	10,12	4,20	2,83	1,36	5,51
SA	22,21	17,02	3,19	1,92	5,60

exp6	bad pixels all	non occl	RMS all	RMS non occl	Prediction Err
GC	9,39	3,61	2,67	1,29	7,91

Fehlerauswertung für Motiv Zickzack

exp1	bad pixels all	non occl	RMS all	RMS non occl	Prediction Err
SAD	24,37	19,31	3,34	2,36	9,70
SSD	23,77	18,78	3,32	2,32	9,71
SAD interval	23,35	18,16	3,26	2,24	9,64
SSD interval	22,64	17,47	3,23	2,21	9,62

exp2	bad pixels all	non occl	RMS all	RMS non occl	Prediction Err
SAD	24,19	19,59	3,27	2,41	10,09
SSD	23,69	19,03	3,24	2,35	9,46
SAD interval	21,13	16,45	3,17	2,23	9,76
SSD interval	20,51	15,88	3,08	2,09	9,35

exp3	bad pixels all	non occl	RMS all	RMS non occl	Prediction Err
DP	15,30	13,49	1,75	1,70	7,83
SO	15,74	15,61	1,83	1,82	5,90
GC	8,34	4,34	2,02	0,92	7,39

exp4	bad pixels all	non occl	RMS all	RMS non occl	Prediction Err
Boxfilter	17,12	11,53	3,13	1,81	7,80
Minfilter	11,48	6,89	2,54	1,24	7,54
Binomialfilter	18,21	12,60	3,31	2,11	8,34
Diffusion	17,43	11,79	3,31	2,10	8,78
Membran	17,43	11,79	3,36	2,17	6,84

exp5	bad pixels all	non occl	RMS all	RMS non occl	Prediction Err
DP	13,51	8,47	2,93	1,74	7,38
SO	14,39	8,51	3,20	1,99	4,96
GC	8,35	2,01	2,60	0,67	5,16
SA	20,09	14,70	3,50	2,48	5,20

exp6	bad pixels all	non occl	RMS all	RMS non occl	Prediction Err
GC	9,05	2,77	2,69	0,96	8,14

Fehlerauswertung für Motiv Glühbirnen

exp1	bad pixels all	non occl	RMS all	RMS non occl	Prediction Err
SAD	60,75	57,89	5,60	5,16	11,58
SSD	58,54	55,57	5,46	4,98	11,42
SAD interval	61,52	58,78	5,71	5,26	11,45
SSD interval	59,21	56,26	5,51	5,05	11,46

exp2	bad pixels all	non occl	RMS all	RMS non occl	Prediction Err
SAD	63,89	61,55	5,55	5,14	11,91
SSD	62,29	59,83	5,50	5,07	11,13
SAD interval	60,48	57,96	5,47	5,07	11,89
SSD interval	60,81	58,32	5,44	5,02	11,13

exp3	bad pixels all	non occl	RMS all	RMS non occl	Prediction Err
DP	56,66	55,80	4,66	4,54	10,21
SO	57,17	56,75	5,22	5,16	7,30
GC	43,86	42,24	3,84	3,46	8,25

exp4	bad pixels all	non occl	RMS all	RMS non occl	Prediction Err
Boxfilter	46,32	42,38	4,81	3,92	8,75
Minfilter	45,73	42,07	4,31	3,78	8,56
Binomialfilter	51,03	47,40	5,06	4,29	9,61
Diffusion	48,95	45,19	4,92	4,08	10,11
Membran	48,95	45,19	4,92	4,08	8,57

exp5	bad pixels all	non occl	RMS all	RMS non occl	Prediction Err
DP	46,19	43,14	4,23	3,56	9,77
SO	47,86	44,27	4,87	4,14	6,41
GC	35,92	31,51	4,19	3,26	6,53
SA	55,90	52,71	5,27	4,60	6,60

exp6	bad pixels all	non occl	RMS all	RMS non occl	Prediction Err
GC	37,13	32,75	4,40	3,58	8,66

Fehlerauswertung für Motiv Würfel

exp1	bad pixels all	non occl	RMS all	RMS non occl	Prediction Err
SAD	29,66	24,03	3,61	2,24	7,58
SSD	28,64	22,88	3,57	2,18	7,63
SAD interval	29,18	23,48	3,60	2,20	7,59
SSD interval	28,78	23,03	3,54	2,15	7,57

exp2	bad pixels all	non occl	RMS all	RMS non occl	Prediction Err
SAD	31,67	26,35	3,64	2,42	7,52
SSD	31,03	25,64	3,63	2,39	7,25
SAD interval	29,86	24,39	3,57	2,32	7,54
SSD interval	29,60	24,09	3,53	2,24	7,25

exp3	bad pixels all	non occl	RMS all	RMS non occl	Prediction Err
DP	18,18	15,89	1,56	1,45	6,47
SO	24,28	23,60	1,69	1,67	5,35
GC	24,37	18,26	1,93	1,59	6,53

exp4	bad pixels all	non occl	RMS all	RMS non occl	Prediction Err
Boxfilter	21,97	15,66	3,46	1,58	6,43
Minfilter	23,61	17,80	3,07	1,40	6,12
Binomialfilter	22,69	16,47	3,60	1,76	6,80
Diffusion	22,12	15,81	3,60	1,77	7,08
Membran	22,19	15,88	3,68	1,93	5,94

exp5	bad pixels all	non occl	RMS all	RMS non occl	Prediction Err
DP	17,06	10,65	3,22	1,29	6,06
SO	17,59	10,93	3,51	1,59	4,37
GC	24,45	18,36	3,28	1,60	4,69
SA	34,26	28,98	3,60	2,23	4,73

exp6	bad pixels all	non occl	RMS all	RMS non occl	Prediction Err
GC	25,02	18,99	3,27	1,49	6,77

Fehlerauswertung für Motiv Jeans

exp1	bad pixels all	non occl	RMS all	RMS non occl	Prediction Err
SAD	10,53	4,33	2,83	1,07	17,75
SSD	10,37	4,05	2,85	1,06	17,56
SAD interval	10,43	4,10	2,83	0,96	17,58
SSD interval	10,47	4,16	2,81	0,95	17,60

exp2	bad pixels all	non occl	RMS all	RMS non occl	Prediction Err
SAD	9,74	3,68	2,64	0,73	17,56
SSD	9,31	3,11	2,64	0,69	17,13
SAD interval	9,62	3,55	2,65	0,71	17,32
SSD interval	9,18	2,99	2,63	0,66	17,06

exp3	bad pixels all	non occl	RMS all	RMS non occl	Prediction Err
DP	6,75	5,30	1,22	1,03	13,66
SO	8,82	7,48	1,77	1,63	9,18
GC	2,86	1,35	1,18	0,81	10,67

exp4	bad pixels all	non occl	RMS all	RMS non occl	Prediction Err
Boxfilter	11,22	4,84	2,92	1,16	15,49
Minfilter	9,56	3,58	2,64	0,71	14,80
Binomialfilter	11,25	4,86	3,00	1,22	16,47
Diffusion	11,24	4,85	3,00	1,25	16,95
Membran	11,36	4,96	3,07	1,35	14,60

exp5	bad pixels all	non occl	RMS all	RMS non occl	Prediction Err
DP	8,23	2,57	2,78	1,14	13,00
SO	8,90	2,43	2,99	1,10	8,68
GC	7,81	1,21	2,82	0,77	8,66
SA	9,10	2,60	2,82	0,88	8,64

exp6	bad pixels all	non occl	RMS all	RMS non occl	Prediction Err
GC	8,65	2,18	2,89	1,11	13,00

Fehlerauswertung für Motiv Pfeil

exp1	bad pixels all	non occl	RMS all	RMS non occl	Prediction Err
SAD	26,26	19,94	4,47	2,62	13,19
SSD	25,23	18,89	4,40	2,58	12,94
SAD interval	26,23	20,00	4,50	2,69	12,89
SSD interval	25,57	19,21	4,43	2,62	12,92

exp2	bad pixels all	non occl	RMS all	RMS non occl	Prediction Err
SAD	30,15	24,47	4,56	2,92	13,25
SSD	29,61	23,82	4,54	2,88	12,44
SAD interval	28,07	22,22	4,54	2,82	12,92
SSD interval	27,69	21,76	4,54	2,83	12,37

exp3	bad pixels all	non occl	RMS all	RMS non occl	Prediction Err
DP	17,36	15,18	1,35	1,21	9,98
SO	23,51	21,29	1,88	1,70	7,85
GC	11,90	11,37	1,45	1,33	8,74

exp4	bad pixels all	non occl	RMS all	RMS non occl	Prediction Err
Boxfilter	19,32	12,47	4,50	2,15	10,13
Minfilter	21,68	15,35	4,11	2,28	9,97
Binomialfilter	21,37	14,56	4,73	2,70	11,04
Diffusion	20,42	13,56	4,74	2,70	11,64
Membran	20,42	13,56	4,80	2,71	9,76

exp5	bad pixels all	non occl	RMS all	RMS non occl	Prediction Err
DP	18,68	12,44	4,21	1,87	9,44
SO	20,44	13,87	4,65	2,46	7,19
GC	16,37	9,13	4,15	1,12	7,26
SA	22,13	15,35	4,50	2,71	7,32

exp6	bad pixels all	non occl	RMS all	RMS non occl	Prediction Err
GC	19,61	12,75	4,63	2,59	9,44

Fehlerauswertung für Motiv Stern

exp1	bad pixels all	non occl	RMS all	RMS non occl	Prediction Err
SAD	28,20	21,96	4,34	3,27	13,96
SSD	25,97	19,61	4,20	3,10	13,81
SAD interval	27,90	21,75	4,33	3,28	13,66
SSD interval	25,84	19,42	4,20	3,10	13,61

exp2	bad pixels all	non occl	RMS all	RMS non occl	Prediction Err
SAD	33,94	28,73	4,86	4,06	13,93
SSD	31,55	26,05	4,73	3,91	12,91
SAD interval	28,60	23,07	4,39	3,39	13,39
SSD interval	28,83	23,32	4,40	3,41	12,70

exp3	bad pixels all	non occl	RMS all	RMS non occl	Prediction Err
DP	21,69	18,24	2,46	2,24	10,40
SO	22,68	21,09	2,93	2,71	9,00
GC	17,56	15,53	2,20	1,47	9,63

exp4	bad pixels all	non occl	RMS all	RMS non occl	Prediction Err
Boxfilter	21,58	14,92	3,96	2,53	11,01
Minfilter	20,76	14,90	3,66	2,56	10,99
Binomialfilter	22,57	15,85	4,02	2,64	12,09
Diffusion	21,41	14,58	3,94	2,48	12,59
Membran	21,41	14,58	3,94	2,48	11,22

exp5	bad pixels all	non occl	RMS all	RMS non occl	Prediction Err
DP	15,13	9,00	3,24	1,31	10,00
SO	15,90	9,00	3,49	1,65	8,34
GC	13,26	5,83	3,30	1,19	8,54
SA	22,16	15,58	4,12	3,01	8,45

exp6	bad pixels all	non occl	RMS all	RMS non occl	Prediction Err
GC	15,07	7,80	3,40	1,61	10,42

Fehlerauswertung für Motiv TigerTest

exp1	bad pixels all	non occl	RMS all	RMS non occl	Prediction Err
SAD	18,03	13,20	2,70	1,92	13,54
SSD	17,45	12,60	2,66	1,87	13,44
SAD interval	17,75	12,91	2,65	1,86	13,49
SSD interval	17,21	12,35	2,61	1,81	13,50

exp2	bad pixels all	non occl	RMS all	RMS non occl	Prediction Err
SAD	21,75	17,58	2,78	2,13	13,41
SSD	20,83	16,61	2,75	2,11	12,85
SAD interval	18,65	14,15	2,65	1,94	13,36
SSD interval	18,57	14,12	2,64	1,95	12,85

exp3	bad pixels all	non occl	RMS all	RMS non occl	Prediction Err
DP	14,56	11,39	1,50	1,47	10,16
SO	18,67	17,46	1,89	1,79	7,30
GC	8,44	6,12	0,91	0,83	8,64

exp4	bad pixels all	non occl	RMS all	RMS non occl	Prediction Err
Boxfilter	13,49	8,35	2,42	1,41	10,61
Minfilter	11,14	6,79	2,13	1,23	10,19
Binomialfilter	13,57	8,44	2,56	1,62	11,42
Diffusion	13,58	8,44	2,56	1,62	11,98
Membran	14,09	9,01	2,64	1,75	9,59

exp5	bad pixels all	non occl	RMS all	RMS non occl	Prediction Err
DP	11,91	7,35	2,44	1,58	9,66
SO	13,74	8,78	2,64	1,77	6,41
GC	7,61	2,26	2,14	0,73	6,52
SA	15,03	10,05	2,57	10,05	6,54

exp6	bad pixels all	non occl	RMS all	RMS non occl	Prediction Err
GC	9,22	4,08	2,37	1,32	9,69

Fehlerauswertung für Motiv Papier

exp1	bad pixels all	non occl	RMS all	RMS non occl	Prediction Err
SAD	53,39	48,87	8,76	8,23	6,84
SSD	52,54	48,06	8,61	8,06	6,89
SAD interval	51,55	47,15	8,65	8,12	6,30
SSD interval	51,71	47,19	8,66	8,12	6,50

exp2	bad pixels all	non occl	RMS all	RMS non occl	Prediction Err
SAD	57,39	53,36	9,01	8,54	6,31
SSD	56,99	52,99	9,17	8,63	5,94
SAD interval	53,53	49,45	9,02	8,50	6,30
SSD interval	55,09	51,03	9,05	8,53	6,00

exp3	bad pixels all	non occl	RMS all	RMS non occl	Prediction Err
DP	27,81	22,33	6,23	6,23	10,36
SO	42,65	39,83	6,90	6,83	4,96
GC	24,00	17,21	6,61	6,62	5,23

exp4	bad pixels all	non occl	RMS all	RMS non occl	Prediction Err
Boxfilter	40,83	34,93	8,38	7,81	5,76
Minfilter	46,19	40,78	8,49	7,91	5,31
Binomialfilter	40,81	34,89	8,60	7,85	6,09
Diffusion	37,53	31,25	8,36	7,56	6,30
Membran	37,53	31,25	8,36	7,56	5,73

exp5	bad pixels all	non occl	RMS all	RMS non occl	Prediction Err
DP	26,43	20,85	7,35	6,52	7,46
SO	26,79	19,39	8,05	7,20	4,51
GC	22,59	14,68	6,72	5,45	4,67
SA	46,07	40,70	7,63	6,68	4,75

exp6	bad pixels all	non occl	RMS all	RMS non occl	Prediction Err
GC	22,28	14,39	7,61	6,63	5,25

Fehlerauswertung für Motiv Stifte

exp1	bad pixels all	non occl	RMS all	RMS non occl	Prediction Err
SAD	30,71	22,89	3,40	2,54	14,15
SSD	28,03	20,33	3,36	2,40	14,19
SAD interval	29,67	22,45	3,40	2,56	14,15
SSD interval	28,74	20,97	3,37	2,45	13,96

exp2	bad pixels all	non occl	RMS all	RMS non occl	Prediction Err
SAD	36,45	29,96	3,48	2,84	14,09
SSD	35,10	28,65	3,40	2,66	13,35
SAD interval	30,98	24,52	3,41	2,66	14,47
SSD interval	32,24	26,16	3,44	2,68	13,49

exp3	bad pixels all	non occl	RMS all	RMS non occl	Prediction Err
DP	22,56	16,99	2,83	2,18	10,71
SO	24,85	22,81	3,51	3,25	9,00
GC	15,15	11,49	3,00	2,54	9,40

exp4	bad pixels all	non occl	RMS all	RMS non occl	Prediction Err
Boxfilter	22,85	14,60	3,43	2,46	11,12
Minfilter	18,41	10,28	3,28	2,39	10,97
Binomialfilter	23,26	14,53	3,40	2,40	12,18
Diffusion	22,74	14,03	3,40	2,40	12,60
Membran	22,77	14,07	3,40	2,35	11,09

exp5	bad pixels all	non occl	RMS all	RMS non occl	Prediction Err
DP	14,14	7,82	2,85	2,10	10,21
SO	17,62	8,61	3,29	2,12	8,46
GC	12,64	2,77	2,97	1,43	8,47
SA	19,77	10,75	3,33	2,08	8,57

exp6	bad pixels all	non occl	RMS all	RMS non occl	Prediction Err
GC	14,50	4,93	3,03	1,62	9,90

Fehlerauswertung für Motiv Teddy

exp1	bad pixels all	non occl	RMS all	RMS non occl	Prediction Err
SAD	18,84	14,48	2,49	2,08	9,67
SSD	17,89	13,40	2,38	1,97	9,68
SAD interval	18,97	14,59	2,50	2,09	9,67
SSD interval	17,62	13,21	2,37	1,95	9,63

exp2	bad pixels all	non occl	RMS all	RMS non occl	Prediction Err
SAD	22,39	18,51	2,69	2,38	9,68
SSD	21,41	17,44	2,62	2,30	9,47
SAD interval	21,18	17,10	2,68	2,33	9,60
SSD interval	20,98	16,96	2,56	2,23	9,48

exp3	bad pixels all	non occl	RMS all	RMS non occl	Prediction Err
DP	10,98	9,07	1,21	1,08	7,41
SO	18,43	17,29	2,31	2,27	6,12
GC	5,26	5,16	1,22	1,08	7,60

exp4	bad pixels all	non occl	RMS all	RMS non occl	Prediction Err
Boxfilter	14,97	10,22	2,39	1,89	8,95
Minfilter	14,67	10,76	2,21	1,83	8,47
Binomialfilter	16,33	11,64	2,48	2,01	9,43
Diffusion	15,49	10,75	2,40	1,90	9,64
Membran	15,49	10,75	2,40	1,90	8,59

exp5	bad pixels all	non occl	RMS all	RMS non occl	Prediction Err
DP	8,16	3,91	1,78	1,05	7,11
SO	8,77	3,95	1,87	1,13	5,36
GC	6,78	1,91	1,78	0,97	5,54
SA	16,77	12,22	2,34	1,84	5,56

exp6	bad pixels all	non occl	RMS all	RMS non occl	Prediction Err
GC	6,72	2,14	1,74	0,96	8,50

Fehlerauswertung für Motiv Chaos flach

exp1	bad pixels all	non occl	RMS all	RMS non occl	Prediction Err
SAD	14,62	7,64	2,64	1,02	7,73
SSD	13,58	6,72	2,65	0,94	7,76
SAD interval	16,12	9,08	2,64	1,05	7,68
SSD interval	14,75	7,72	2,63	0,97	7,63

exp2	bad pixels all	non occl	RMS all	RMS non occl	Prediction Err
SAD	15,12	8,92	2,63	1,07	8,14
SSD	13,80	7,28	2,62	0,96	7,59
SAD interval	14,86	8,48	2,58	0,98	7,80
SSD interval	14,58	8,07	2,57	0,95	7,51

exp3	bad pixels all	non occl	RMS all	RMS non occl	Prediction Err
DP	10,25	7,10	1,29	0,97	6,53
SO	16,62	14,47	1,78	1,52	5,89
GC	7,31	3,43	1,28	0,86	7,01

exp4	bad pixels all	non occl	RMS all	RMS non occl	Prediction Err
Boxfilter	16,28	9,08	2,94	1,20	7,00
Minfilter	10,81	4,94	2,53	0,97	6,72
Binomialfilter	16,36	9,09	3,05	1,30	7,71
Diffusion	16,38	9,11	3,06	1,31	8,05
Membran	16,69	9,41	3,15	1,44	7,39

exp5	bad pixels all	non occl	RMS all	RMS non occl	Prediction Err
DP	9,85	4,05	2,86	1,02	6,23
SO	12,02	4,82	2,91	1,17	5,06
GC	9,31	1,80	2,72	0,84	5,26
SA	12,82	5,41	2,75	1,18	5,29

exp6	bad pixels all	non occl	RMS all	RMS non occl	Prediction Err
GC	8,72	1,84	2,61	0,76	7,27

Fehlerauswertung für Motiv Chaos tief

exp1	bad pixels all	non occl	RMS all	RMS non occl	Prediction Err
SAD	16,16	9,97	2,80	1,34	10,34
SSD	15,33	9,32	2,80	1,27	10,34
SAD interval	16,52	10,33	2,78	1,32	10,18
SSD interval	15,61	9,48	2,76	1,30	10,27

exp2	bad pixels all	non occl	RMS all	RMS non occl	Prediction Err
SAD	15,08	9,48	2,73	1,31	10,69
SSD	14,67	8,82	2,69	1,24	9,78
SAD interval	13,87	8,20	2,66	1,24	10,25
SSD interval	13,77	7,94	2,65	1,19	9,65

exp3	bad pixels all	non occl	RMS all	RMS non occl	Prediction Err
DP	10,97	8,86	1,38	1,17	8,64
SO	20,77	17,44	1,82	1,75	6,64
GC	7,57	3,52	1,05	0,85	7,95

exp4	bad pixels all	non occl	RMS all	RMS non occl	Prediction Err
Boxfilter	18,09	11,66	3,11	1,62	8,40
Minfilter	12,15	6,80	2,60	1,14	8,14
Binomialfilter	17,75	11,30	3,23	1,74	9,02
Diffusion	17,75	11,30	3,23	1,76	9,49
Membran	17,85	11,41	3,32	1,87	7,99

exp5	bad pixels all	non occl	RMS all	RMS non occl	Prediction Err
DP	14,10	8,52	2,90	1,33	7,98
SO	15,42	8,94	3,16	1,64	5,92
GC	10,42	3,58	2,82	1,06	6,06
SA	17,01	10,75	3,03	1,50	6,08

exp6	bad pixels all	non occl	RMS all	RMS non occl	Prediction Err
GC	9,36	2,95	2,84	0,95	8,46

A.4. Endergebnis der Fehlerauswertung

Durchschnittswerte aller Motive

exp1	bad pixels all	non occl	RMS all	RMS non occl	Prediction Err
SAD	29,32	23,79	4,03	2,97	11,93
SSD	28,30	22,74	3,97	2,88	11,86
SAD interval	28,95	23,46	4,00	2,93	11,80
SSD interval	28,15	22,56	3,95	2,86	11,80

exp2	bad pixels all	non occl	RMS all	RMS non occl	Prediction Err
SAD	31,31	26,37	4,08	3,13	11,94
SSD	30,44	25,40	4,05	3,06	11,33
SAD interval	28,74	23,63	4,00	2,99	11,78
SSD interval	28,81	23,71	3,97	2,95	11,31

exp3	bad pixels all	non occl	RMS all	RMS non occl	Prediction Err
DP	20,12	17,36	2,29	2,06	9,62
SO	25,66	23,99	2,86	2,72	7,28
GC	14,00	10,84	2,21	1,73	8,29

exp4	bad pixels all	non occl	RMS all	RMS non occl	Prediction Err
Boxfilter	22,69	16,55	3,83	2,49	9,73
Minfilter	21,09	15,67	3,50	2,35	9,41
Binomialfilter	23,65	17,50	3,97	2,67	10,48
Diffusion	22,70	16,48	3,92	2,60	10,95
Membran	22,77	16,55	3,95	2,64	9,28

exp5	bad pixels all	non occl	RMS all	RMS non occl	Prediction Err
DP	18,32	12,86	3,40	2,04	8,94
SO	19,67	13,36	3,76	2,35	6,57
GC	14,52	7,71	3,29	1,54	6,69
SA	26,12	20,25	3,96	3,27	6,75

exp6	bad pixels all	non occl	RMS all	RMS non occl	Prediction Err
GC	15,38	8,80	3,44	1,91	9,06

A.5. Beste Ergebnisse aus einer Serie

Statistik der Anzahl der besten Ergebnisse in Abhängigkeit von den Parametern

Experiment 1

Parameter	bad pixels all	non occl	RMS all	RMS non occl	Prediction Err
SAD09	4	7	2	4	5
SAD09t50	11	9	10	11	11
SAD09t20	1	1	3	1	
SAD09t10			1		
SAD09t05			1	1	
SAD09t02					
SAD09t01					

Parameter	bad pixels all	non occl	RMS all	RMS non occl	Prediction Err
SSD09	2	2	1	2	2
SSD09t50	8	9	6	7	11
SSD09t20	6	5	9	6	3
SSD09t10					
SSD09t05					
SSD09t02			1	1	
SSD09t01					

Parameter	bad pixels all	non occl	RMS all	RMS non occl	Prediction Err
SAD09b	2	3	3	4	2
SAD09bt50	12	11	8	9	14
SAD09bt20	1	1	5	2	
SAD09bt10	1	1	2	1	
SAD09bt05					
SAD09bt02			1	1	
SAD09bt01					

Parameter	bad pixels all	non occl	RMS all	RMS non occl	Prediction Err
SSD09b					2
SSD09bt50	8	8	5	7	9
SSD09bt20	6	8	7	9	6
SSD09bt10	2	1	3	1	
SSD09bt05					
SSD09bt02			1	1	
SSD09bt01					

Experiment 2

Parameter	bad pixels all	non occl	RMS all	RMS non occl	Prediction Err
SADmf09	14	14	9	11	16
SADmf09t50	1	1	6	3	
SADmf09t20	1	1	1		
SADmf09t10			1	1	
SADmf09t05					
SADmf09t02			1	1	
SADmf09t01					

Parameter	bad pixels all	non occl	RMS all	RMS non occl	Prediction Err
SSDmf09	6	11	7	10	16
SSDmf09t50	7	3	7	6	
SSDmf09t20	2	1	3	2	
SSDmf09t10	1	1			
SSDmf09t05			1	1	
SSDmf09t02					
SSDmf09t01					

Parameter	bad pixels all	non occl	RMS all	RMS non occl	Prediction Err
SADmf09b	6	9	6	8	16
SADmf09bt50	5	2	5	4	
SADmf09bt20	1		1		
SADmf09bt10	1	2	2	1	
SADmf09bt05	1	1	3	4	
SADmf09bt02	2	2	1	2	
SADmf09bt01					

Parameter	bad pixels all	non occl	RMS all	RMS non occl	Prediction Err
SSDmf09b	2	4	4	6	16
SSDmf09bt50	6	6	5	7	
SSDmf09bt20	3	2	5		
SSDmf09bt10			2	1	
SSDmf09bt05	1	1	2	2	
SSDmf09bt02	3	3	2	2	
SSDmf09bt01	1	1		1	

Experiment 3

Parameter	bad pixels all	non occl	RMS all	RMS non occl	Prediction Err
DPm1o20s0020					8
DPm1o20s0020t50					
DPm1o20s0020t20					
DPm1o20s0020t10					
DPm1o20s0020t05					
DPm1o20s0020t02					
DPm1o20s0020b					6
DPm1o20s0020bt50	2	4	1	2	2
DPm1o20s0020bt20	9	8	7	7	
DPm1o20s0020bt10	4	4	7	6	
DPm1o20s0020bt05	1		1	1	
DPm1o20s0020bt02					

Parameter	bad pixels all	non occl	RMS all	RMS non occl	Prediction Err
S0m1s0020					16
S0m1s0020t50					
S0m1s0020t20					
S0m1s0020t10					
S0m1s0020t05					
S0m1s0020t02					
S0m1s0020b					
S0m1s0020bt50					
S0m1s0020bt20					
S0m1s0020bt10	2	3	2	3	
S0m1s0020bt05	11	12	7	7	
S0m1s0020bt02	3	1	7	6	

Parameter	bad pixels all	non occl	RMS all	RMS non occl	Prediction Err
GCm1s0020					16
GCm1s0020t50		1			
GCm1s0020t20	4	4	2	4	
GCm1s0020t10	1	1	1		
GCm1s0020t05			1	1	
GCm1s0020b					
GCm1s0020bt50	1	1			
GCm1s0020bt20		3		2	
GCm1s0020bt10	10	6	12	9	
GCm1s0020bt05			1		

Experiment 4

Parameter	bad pixels all	non occl	RMS all	RMS non occl	Prediction Err
SAD03			1	1	16
SAD05					
SAD07			2	1	
SAD09	1	1	1	1	
SAD11					
SAD13	2	2	1	2	
SAD15			2	2	
SAD17	1	1	2	2	
SAD19			1	1	
SAD21			1	2	
SAD23	3	4	2	2	
SAD25	2	2	2		
SAD27			3	2	
SAD29	7	7	5	5	

Parameter	bad pixels all	non occl	RMS all	RMS non occl	Prediction Err
SADmf03			1	1	16
SADmf05					
SADmf07					
SADmf09			1		
SADmf11		1	1	1	
SADmf13	1			1	
SADmf15	1	1	2	2	
SADmf17			3		
SADmf19	1	1	4	1	
SADmf21					
SADmf23	2	2	1	1	
SADmf25	1	2	4	3	
SADmf27	4	3	1	2	
SADmf29	6	6	8	7	

Parameter	bad pixels all	non occl	RMS all	RMS non occl	Prediction Err
SADbf02					16
SADbf04			2	1	
SADbf06	1	1	1	1	
SADbf08				1	
SADbf10			1		
SADbf12	2	2	1		
SADbf14			2		
SADbf16			2	1	
SADbf18			5	2	
SADbf20			4	1	
SADbf22			1	1	
SADbf24	1	1	1	2	
SADbf26	1	1	1	2	
SADbf28	12	12	7	8	

Parameter	bad pixels all	non occl	RMS all	RMS non occl	Prediction Err
SADdiff010			2		16
SADdiff020	1	1	1	2	
SADdiff030			1	1	
SADdiff040	2	2	2		
SADdiff050			3	3	
SADdiff060			4	2	
SADdiff070			2	3	
SADdiff080		1		2	
SADdiff090	1		1	1	
SADdiff100			1	1	
SADdiff110			1	1	
SADdiff120					
SADdiff130	1	1			
SADdiff140	1	1			
SADdiff150	10	10	6	6	

Parameter	bad pixels all	non occl	RMS all	RMS non occl	Prediction Err
SADmemb09					16
SADmemb08					
SADmemb07					
SADmemb06			1		
SADmemb05					
SADmemb04			1	2	
SADmemb03			1	1	
SADmemb02	1	1	1	1	
SADmemb01	2	2	5	5	
SADmemb00	13	13	8	9	

Experiment 5

Parameter	bad pixels all	non occl	RMS all	RMS non occl	Prediction Err
DPm1o20s0005			1		10
DPm1o20s0010					
DPm1o20s0020					
DPm1o20s0050					
DPm1o20s0100					
DPm1o20s0200	2	2			
DPm1o20s0500	2	2			
DPm1o20s1000					1
DPm1o50s0005			5	3	5
DPm1o50s0010			5	1	
DPm1o50s0020					
DPm1o50s0050	2	2		1	
DPm1o50s0100		1			
DPm1o50s0200	3	2			
DPm1o50s0500	2	2			
DPm1o50s1000					
DPm1o80s0005			9	7	
DPm1o80s0010			3	1	
DPm1o80s0020					
DPm1o80s0050	1	1			
DPm1o80s0100		1		1	
DPm1o80s0200	1	1		1	
DPm1o80s0500	2	2	1	2	
DPm1o80s1000	1	1			

Parameter	bad pixels all	non occl	RMS all	RMS non occl	Prediction Err
S0m1s0005					14
S0m1s0010					2
S0m1s0020					
S0m1s0050					
S0m1s0100			2	3	
S0m1s0200	6	6	4	4	
S0m1s0500	6	6	8	8	
S0m1s1000	4	4	2	2	

Parameter	bad pixels all	non occl	RMS all	RMS non occl	Prediction Err
GCm1s0005					16
GCm1s0010					
GCm1s0020	1	1	1		
GCm1s0050	2	2	5	3	
GCm1s0100	3	3	5	5	
GCm1s0200	7	7	3	2	
GCm1s0500	3	3	3	6	
GCm1s1000					

Parameter	bad pixels all	non occl	RMS all	RMS non occl	Prediction Err
SAm1s0005					16
SAm1s0010					
SAm1s0020					
SAm1s0050			4		
SAm1s0100			6	2	
SAm1s0200	4	4	4	5	
SAm1s0500	11	11	4	8	
SAm1s1000	1	1		2	

Experiment 6

Parameter	bad pixels all	non occl	RMS all	RMS non occl	Prediction Err
GCm1s0020b					16
GCm1s0020bt08p2					
GCm1s0020bt08p4	1	1	1	1	
GCm1s0050b	1	1	4	3	
GCm1s0050bt08p2	2	2	4	5	
GCm1s0050bt08p4	12	12	7	9	

A.6. Benötigte Zeit zur Erstellung einer Tiefenkarte

Experiment	Verfahren	Zeit [sec]
exp1	Boxfilter	0,8 – 1
exp2	Minfilter	1,2 – 1,6
exp3	DP SO GC	1,5 – 1,7 1,6 – 1,9 800 – 1700
exp4	Boxfilter Minfilter Binomialfilter Diffusion Membran	0,8 – 0,9 1,35 – 1,65 2 – 28 (pro Iteration ca 2 s) bis 19 (pro 10 Iterationen 1,2s) 35
exp5	DP SO GC SA	1,45 – 1,55 1,65 – 1,75 600 – 1300 650 – 700 (bei 500 Iterationen)
exp6	GC	900 – 1200

Alle Zeiten gelten für eine Taktfrequenz von 1 GHz auf einem Pentium III PC für Bilder mit 450 * 300 Pixeln und d_{\max} von 40 – 45

A.8. Lexikon der Abkürzungen

d	Disparität
disp_min	kleinste Disparität in einem Stereobildpaar
disp_max	größte Disparität in einem Stereobildpaar
disp_scale	Skalierung der Disparitäten bei Umwandlung in Helligkeitswerte
disp_step	Abstufung der Disparitäten
DP	Dynamic Programming
DSI	Disparity Space Image (Disparitäts-Raum-Bild)
GC	Graph Cuts
H	Helligkeitswert (0 = schwarz, 255 = weiß)
RMS	Root Mean Square (Fehlermaß, in das die Wurzel der quadrierten Differenz der entsprechenden Pixel zweier Bilder eingeht)
SA	Simulated Annealing
SAD	Sum of Absolute Differences (Anhäufungsmethode)
SO	Scanline Optimization
SSD	Sum of Squared Differences (Anhäufungsmethode)
WTA	Winner-Take-All (Verfahren zur Bestimmung der Disparitäten in lokalen Methoden)

A.9. Literaturverzeichnis

- [1] Daniel Scharstein, Richard Szeliski
"A Taxonomy and Evaluation of Dense Two-Frame Stereo Correspondence Algorithms"
www.middlebury.edu/stereo 6.10.2004
- [2] www.avglab.com/andrew/soft 6.10.2004
- [3] Hermann Schichl
"Numerik 2"
www.mat.univie.ac.at/~herman/skripten/numerik2.pdf 31.1.2005
- [4] www.loehnertz.de/Diplomarbeit/html/node3 22.10.2004
- [5] wwwai.wu.wien.ac.at/~hahsler/PPAP/projekte/SS2001/Kammlander/doku/projekt/node8
22.10.2004
- [6] www.advanced-planning.de/advancedplanning-177 22.10.2004
- [7] www.www-sop.inria.fr/robotvis/personnel/fabad/PhD/GraphCut 22.10.2004
- [8] www.de.wikipedia.org/wiki/Computer_Vision 24.10.2004
- [9] Christian Thiel
"Verdeckungsanalyse in Stereobildern"
www.informatik.uni-mannheim.de/pi4/lectures/ws0203/seminar_data/ChristianThiel.pdf
6.2.2005
- [10] Yuri Boykov, Vladimir Kolmogorov
"An Experimental Comparison of Min-Cut/Max-Flow Algorithms for Energy
Minimization in Vision"
www.cs.cornell.edu/~rdz/graphcuts 22.10.04
- [11] Yuri Boykov, Olga Veksler, Ramin Zabih
"A New Algorithm for Energy Minimization with Discontinuities"
www.dblp.uni-trier.de/db/indices/a-tree/b/Boykov:Yuri 22.10.04

- [12] Stan Birchfeld, Carlo Tomasi
"Depth Discontinuities by Pixel-to-Pixel Stereo"
www.bj.middlebury.edu/schar/stereo/web/results.php 6.2.2005
- [13] Yuri Boykov, Olga Veksler, Ramin Zabih
"Fast Approximate Energy Minimization via Graph Cuts"
www.bj.middlebury.edu/schar/stereo/web/results.php 6.2.2005
- [14] Horst Kuchling
"Taschenbuch der Physik"
ISBN 3-343-00884-2
- [15] Stiftung Warentest
Sonderheft Foto
November 1989
- [16] www.rsb.info.nih.gov/ij/ 6.2.2005
- [17] Werner Bailer
"Writing ImageJ Plugins – A Tutorial"
www.mtd.fh-hagenberg.at/depot/imaging/imagej 6.2.2005
- [18] Dietmar Kunz
Skript zu Digitale Bildverarbeitung "17. Bewegung"
www.fo.fh-koeln.de/dozenten/dkunz/skripte/bv/bv17.pdf 6.2.2005
- [19] Richard Szeliski
"Prediction Error as a Quality Metric for Motion and Stereo"
www.research.microsoft.com/vision/visionbasedmodeling/publications/szeliski-ICCV99.pdf
31.3.05
- [20] Triggs, Zisserman, Szeliski
"Vision Algorithms: Theory and Practice" S. 1 – 19:
Szeliski, Zabih: "An Experimental Comparison of Stereo Algorithms"
Springer Verlag, ISSN 0302 – 9743

- [21] Daniel Scharstein, Richard Szeliski
"Stereo Matching with Non-Linear Diffusion"
www.research.microsoft.com/vision/visionbasedmodeling/publications/ScharsteinSzeliski-CVPR96.pdf
31.3.05
- [22] Kanade, Okutomi
"A Stereo Matching Algorithm with an Adaptive Window"
www.informatik.uni-trier.de/~ley/db/journals/pami/pami16.html#Kanade094
31.3.05
- [23] Bernd Jähne
"Digitale Bildverarbeitung"
Springer Verlag ISBN 3-540-41260-3
- [24] Harris Sunyoto, Wannes van der Mark, Dariu M. Gavrilă
"A Comparative Study of Fast Dense Stereo Vision Algorithms"
www.gavrila.com/iv04_stereo.pdf 11.2.05
- [25] www.research.microsoft.com/users/szeliski/publications 31.3.05
- [26] www.de.wikipedia.org/wiki/Algorithmus 4.4.05

A.10. Abbildungsverzeichnis

- [A1] Abb. 2: S. S. Intille, A. F. Bobick:
Disparity-Space-Images and Large Occlusion Stereo
MIT Media Lab Perceptual Computing Group Technical Report No. 220
Massachusetts Institute of Technology
- [A2] Abb. 3a: Stereobild Corridor
www-dbv.cs.uni-bonn.de/stereo_data 31.3.05
- [A3] Abb. 6a, 6b: www-sop.inria.fr/robotvis/personnel/fabad/PhD/GraphCut 22.10.04
- [A4] Abb. 12: Dietmar Kunz, Skript zu Digitale Bildverarbeitung "17. Bewegung"
www.fo.fh-koeln.de/dozenten/dkunz/skripte/bv/bv17.pdf 6.2.2005

[A5] Abb. 13a: isgwww.cs.uni-magdeburg.de/~karsten/3dcv 31.3.05

[A6] Abb. 5b und Abb. 37:

www.fh-augsburg.de/informatik/projekte/mebib/emiel/entw_inf/or_verf/2d_vis 4.4.05

[A7] Abb. 5a: www.advanced-planning.de/advancedplanning-153 4.4.05

[A8] Abb. 36: Bernd Jähne

"Digitale Bildverarbeitung"

Springer Verlag ISBN 3-540-41260-3

[A9] Eigenes Archiv:

Abb. 1a, 1b, 3b, 3c, 4a-d, 7a-d, 8a-c, 9a, 9b, 10a-d, 11a-d, 13b, 14, 15, 16, 17, 18, 19, 20, 21,
22, 23, 24, 25, 26, 27a, 27b, 28, 29, 30a, 30b, 31a-d, 32a-d, 33a-d, 34, 35a, 35b

A.11. Inhalte der CDs

CD 1 beinhaltet die Ergebnisse aus 16 selbst erstellten Stereobildern.

CD 2 enthält Ergebnisse der Motive von Scharstein und Szeliski, Beispiele für Fehlerkarten, x-y-Schnitte des DSI, Subpixelverfeinerung und Prediction Error, sowie Zwischenergebnisse des Prozesses des Simulated Annealing.

Auf der CD befinden sich ebenfalls die originalen (unentzerrten) Stereobildpaare, Quellcode des Stereoprogramms, Quellcode des Entzerrungsprogramms und Literatur zum Thema.

Eidesstattliche Erklärung:

Ich versichere hiermit, die vorgelegte Arbeit in dem gemeldeten Zeitraum ohne fremde Hilfe verfasst und mich keiner anderen als der angegebenen Hilfsmittel und Quellen bedient zu haben.

Köln, den 10.5.2005

Sylvia Lange

Sperrvermerk:

Die vorgelegte Arbeit unterliegt keinem Sperrvermerk.

Weitergabeerklärung:

Ich erkläre hiermit mein Einverständnis, dass das vorliegende Exemplar meiner Diplomarbeit oder eine Kopie hiervon für wissenschaftliche Zwecke verwendet werden darf.

Köln, den 10.5.2005

Sylvia Lange